

Maosong Sun Min Zhang
Dekang Lin Haifeng Wang (Eds.)

LNAI 8202

Chinese Computational Linguistics *and* Natural Language Processing Based on Naturally Annotated Big Data

12th China National Conference, CCL 2013 *and*
First International Symposium, NLP-NABD 2013
Suzhou, China, October 2013, Proceedings



 Springer

Lecture Notes in Artificial Intelligence 8202

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Maosong Sun Min Zhang Dekang Lin
Haifeng Wang (Eds.)

Chinese Computational
Linguistics *and*
Natural Language Processing
Based on Naturally Annotated
Big Data

12th China National Conference, CCL 2013 *and*
First International Symposium, NLP-NABD 2013
Suzhou, China, October 10-12, 2013
Proceedings

Volume Editors

Maosong Sun
Tsinghua University, Department of Computer Science and Technology
Beijing, China
E-mail: sms@tsinghua.edu.cn

Min Zhang
Soochow University, School of Computer Science and Technology
Suzhou, China
E-mail: minzhang@suda.edu.cn

Dekang Lin
Google Inc., Mountain View, CA, USA
E-mail: lindek@google.com

Haifeng Wang
Baidu Inc., Beijing, China
E-mail: wanghaifeng@baidu.com

ISSN 0302-9743
ISBN 978-3-642-41490-9
DOI 10.1007/978-3-642-41491-6
Springer Heidelberg New York Dordrecht London

e-ISSN 1611-3349
e-ISBN 978-3-642-41491-6

Library of Congress Control Number: 2013950033

CR Subject Classification (1998): I.2.7, I.2, H.3, I.7, H.2

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2013

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

Welcome to the proceedings of the 12th China National Conference on Computational Linguistics (12th CCL) and the First International Symposium on Natural Language Processing Based on Naturally Annotated Big Data (1st NLP-NABD). The conference was hosted by Soochow University.

CCL is a bi-annual conference that started in 1991. It is the flagship conference of the Chinese Information Processing Society (CIPS), which is the largest NLP scholar and expert community in China. CCL is a premier nation-wide forum for disseminating new scholarly and technological work in computational linguistics, with a major emphasis on computer processing of the languages in China such as Mandarin, Tibetan, Mongolian, and Uyghur.

Affiliated with the 12th CCL, The First International Symposium on Natural Language Processing Based on Naturally Annotated Big Data (NLP-NABD) covered all the NLP topics, with particular focus on methodologies and techniques relating to naturally annotated big data. In contrast to manually annotated data such as treebanks that are constructed for specific NLP tasks, naturally annotated data come into existence through users' normal activities, such as writing, conversation, and interactions on the Web. Although the original purposes of these data typically were unrelated to NLP, they can nonetheless be purposefully exploited by computational linguists to acquire linguistic knowledge. For example, punctuation marks in Chinese text can help word boundaries identification, social tags in social media can provide signals for keyword extraction, categories listed in Wikipedia can benefit text classification. The natural annotation can be explicit, as in the above examples, or implicit, as in Hearst patterns (e.g., "Beijing and other cities" implies "Beijing is a city"). This symposium focuses on numerous research challenges ranging from very-large-scale unsupervised/semi-supervised machine learning (deep learning, for instance) of naturally annotated big data to integration of the learned resources and models with existing handcrafted "core" resources and "core" language computing models.

The Program Committee selected 127 papers (95 Chinese papers and 32 English papers) out of 252 submissions from China, Hong Kong (region), Japan, and Poland for publication. The English papers cover the following topics:

- Word segmentation (7)
- Sentiment analysis, opinion mining and text classification (4)
- Text mining, open-domain information extraction and machine reading of the Web (3)
- Statistical and machine learning methods in NLP (3)
- Machine translation (3)
- Tagging and Chunking (2)
- Language resources and annotation (2)

- Discourse, coreference and pragmatics (2)
- Speech recognition and synthesis (2)
- Lexical semantics and ontologies (1)
- Semantics (1)
- Large-scale knowledge acquisition and reasoning (1)
- Open-domain question answering (1)

The final program for the 12th CCL and the First NLP-NABD was the result of a great deal of work by many dedicated colleagues. We want to thank, first of all, the authors who submitted their papers, and thus contributed to the creation of the high-quality program that allowed us to look forward to an exciting joint conference. We are deeply indebted to all the Program Committee members for providing high-quality and insightful reviews under a tight schedule. We are extremely grateful to the sponsors of the conference. Finally, we extend a special word of thanks to all the colleagues of the Organizing Committee and Secretariat for their hard work in organizing the conference, and to Springer for their assistance in publishing the proceedings in due time.

On behalf of the program and organizing committees, we hope all we have done will make the conference successful, and make it interesting for all the participants. We also believe that your visit to Suzhou, a famous and beautiful historical and cultural city in China, will be a really valuable memory.

Maosong Sun (CCL Program Committee Chair)
Ting Liu, Le Sun, and Min Zhang (CCL Program Committee Co-Chairs)
Maosong Sun, Dekang Lin, and Haifeng Wang (NLP-NABD Program
Committee Chairs)

Organization

General Chairs

Bo Zhang	Tsinghua University, China
Haoming Zhang	Ministry of Education, China
Zhendong Dong	Hownet, China

Program Committee

12th CCL Program Chair

Maosong Sun	Tsinghua University, China
-------------	----------------------------

12th CCL Program Co-chairs

Ting Liu	Harbin Institute of Technology, China
Le Sun	Institute of Software, Chinese Academy of Sciences, China
Min Zhang	Soochow University, China

12th CCL Program Committee

Dongfeng Cai	Shenyang Aerospace University, China
Baobao Chang	Peking University, China
Qunxiu Chen	Tsinghua University, China
Xiaohe Chen	Nanjing Normal University, China
Xueqi Cheng	Institute of Computing Technology, Chinese Academy of Sciences, China
Key-Sun Choi	KAIST, Korea
Li Deng	Microsoft Research, USA
Alexander Gelbukh	National Polytechnic Institute, Mexico
Josef van Genabith	Dublin City University, Ireland
Randy Goebel	University of Alberta, Canada
Tingting He	Huazhong Normal University, China
Isahara Hitoshi	Toyohashi University of Technology, Japan
Heyan Huang	Beijing Polytechnic University, China
Xuanjing Huang	Fudan University, China
Donghong Ji	Wuhan University, China
Turgen Ibrahim	Xinjiang University, China
Shiyong Kang	Ludong University, China
Sadao Kurohashi	Kyoto University, Japan
Kiong Lee	ISO TC37, Korea
Hang Li	Huawei, Hong Kong, SAR China
Ru Li	Shanxi University, China

VIII Organization

Dekang Lin	Google, USA
Qun Liu	Institute of Computing Technology, Chinese Academy of Sciences, China
Shaoming Liu	Fuji Xerox, Japan
Qin Lu	Polytechnic University of Hong Kong, Hong Kong, SAR China
Wolfgang Menzel	University of Hamburg, Germany
Jian-Yun Nie	University of Montreal, Canada
Yanqiu Shao	Beijing Language and Culture University, China
Xiaodong Shi	Xiamen University, China
Rou Song	Beijing Language and Culture University, China
Jian Su	Institute for Infocomm Research, Singapore
Benjamin Ka Yin Tsou	The Hong Kong Institute of Education, Hong Kong, SAR China
Haifeng Wang	Baidu, China
Fei Xia	University of Washington, USA
Feiyu Xu	DFKI, Germany
Nianwen Xue	Brandeis University, USA
Ping Xue	Research & Technology, The Boeing Company
Erhong Yang	Beijing Language and Culture University, China
Tianfang Yao	Shanghai Jiaotong University, China
Shiwen Yu	Peking University, China
Quan Zhang	Institute of Acoustics, Chinese Academy of Sciences, China
Jun Zhao	Institute of Automation, Chinese Academy of Sciences, China
Guodong Zhou	Soochow University, China
Ming Zhou	Microsoft Research Asia, China
Jingbo Zhu	Northeast University, China

First NLP-NABD Program Chairs

Maosong Sun	Tsinghua University, China
Dekang Lin	Google, USA
Haifeng Wang	Baidu, China

First NLP-NABD Program Committee

Key-Sun Choi	KAIST, Korea
Li Deng	Microsoft Research, USA
Alexander Gelbukh	National Polytechnic Institute, Mexico
Josef van Genabith	Dublin City University, Ireland
Randy Goebel	University of Alberta, Canada
Isahara Hitoshi	Toyohashi University of Technology, Japan

Xuanjing Huang	Fudan University, China
Donghong Ji	Wuhan University, China
Sadao Kurohashi	Kyoto University, Japan
Kiong Lee	ISO TC37, Korea
Hang Li	Huawei, Hong Kong
Hongfei Lin	Dalian Polytechnic University, China
Qun Liu	Institute of Computing, Chinese Academy of Sciences, China
Shaoming Liu	Fuji Xerox, Japan
Ting Liu	Harbin Institute of Technology, China
Yang Liu	Tsinghua University, China
Qin Lu	Polytechnic University of Hong Kong, Hong Kong, SAR China
Wolfgang Menzel	University of Hamburg, Germany
Hwee Tou Ng	National University of Singapore, Singapore
Jian-Yun Nie	University of Montreal, Canada
Jian Su	Institute for Infocomm Research, Singapore
Zhifang Sui	Peking University, China
Le Sun	Institute of Software, Chinese Academy of Sciences, China
Benjamin Ka Yin Tsou	The Hong Kong Institute of Education, Hong Kong, SAR China
Fei Xia	University of Washington, USA
Feiyu Xu	DFKI, Germany
Nianwen Xue	Brandeis University, USA
Ping Xue	Research & Technology, The Boeing Company
Jun Zhao	Institute of Automation, Chinese Academy of Sciences, China
Guodong Zhou	Soochow University, China
Ming Zhou	Microsoft Research Asia, China

Organizing Committee

Organizing Committee Chair

Qiaoming Zhu	Soochow University, China
--------------	---------------------------

Organizing Committee Co-chair

Yang Liu	Tsinghua University, China
Longhua Qian	Soochow University, China

Table of Contents

Word Segmentation

Improving Chinese Word Segmentation Using Partially Annotated Sentences	1
<i>Kaixu Zhang, Jinsong Su, and Changle Zhou</i>	
Chinese Natural Chunk Research Based on Natural Annotations in Massive Scale Corpora: Exploring Work on Natural Chunk Recognition Using Explicit Boundary Indicators	13
<i>Zhi-e Huang, En-dong Xun, Gao-qi Rao, and Dong Yu</i>	
A Kalman Filter Based Human-Computer Interactive Word Segmentation System for Ancient Chinese Texts	25
<i>Tongfei Chen, Weimeng Zhu, Xueqiang Lv, and Junfeng Hu</i>	
Chinese Word Segmentation with Character Abstraction	36
<i>Le Tian, Xipeng Qiu, and Xuanjing Huang</i>	
A Refined HDP-Based Model for Unsupervised Chinese Word Segmentation	44
<i>Wenzhe Pei, Dongxu Han, and Baobao Chang</i>	
Enhancing Chinese Word Segmentation with Character Clustering	52
<i>Yijia Liu, Wanxiang Che, and Ting Liu</i>	
Integrating Multi-source Bilingual Information for Chinese Word Segmentation in Statistical Machine Translation	61
<i>Wei Chen, Wei Wei, Zhenbiao Chen, and Bo Xu</i>	

Open-Domain Q&A

Interactive Question Answering Based on FAQ	73
<i>Song Liu, Yi-Xin Zhong, and Fu-Ji Ren</i>	

Discourse, Coreference and Pragmatics

Document Oriented Gap Filling of Definite Null Instantiation in FrameNet	85
<i>Ning Wang, Ru Li, Zhangzhang Lei, Zhiqiang Wang, and Jingpan Jin</i>	
Interesting Linguistic Features in Coreference Annotation of an Inflectional Language	97
<i>Maciej Ogrodniczuk, Katarzyna Głowińska, Mateusz Kopeć, Agata Savary, and Magdalena Zawistawska</i>	

Statistical and Machine Learning Methods in NLP

Semi-supervised Learning with Transfer Learning	109
<i>Huiwei Zhou, Yan Zhang, Degen Huang, and Lishuang Li</i>	
Online Distributed Passive-Aggressive Algorithm for Structured Learning	120
<i>Jiayi Zhao, Xipeng Qiu, Zhao Liu, and Xuanjing Huang</i>	
Power Law for Text Categorization	131
<i>Wuying Liu, Lin Wang, and Mianzhu Yi</i>	

Semantics

Natural Language Understanding for Grading Essay Questions in Persian Language	144
<i>Iman Mokhtari-Fard</i>	

Text Mining, Open-Domain Information Extraction and Machine Reading of the Web

Learning to Extract Attribute Values from a Search Engine with Few Examples	154
<i>Xingxing Zhang, Tao Ge, and Zhifang Sui</i>	
User-Characteristics Topic Model	166
<i>Wenfeng Li, Xiaojie Wang, and Shaowei Jiang</i>	
Mining User Preferences for Recommendation: A Competition Perspective	179
<i>Shaowei Jiang, Xiaojie Wang, Caixia Yuan, and Wenfeng Li</i>	

Sentiment Analysis, Opinion Mining and Text Classification

A Classification-Based Approach for Implicit Feature Identification	190
<i>Lingwei Zeng and Fang Li</i>	
Role of Emoticons in Sentence-Level Sentiment Classification	203
<i>Martin Min, Tanya Lee, and Ray Hsu</i>	
Emotional McGurk Effect? A Cross-Cultural Investigation on Emotion Expression under Vocal and Facial Conflict	214
<i>Aijun Li, Qiang Fang, Yuan Jia, and Jianwu Dang</i>	
Pests Hidden in Your Fans: An Effective Approach for Opinion Leader Discovery	227
<i>Binyang Li, Kam-fai Wong, Lanjun Zhou, Zhongyu Wei, and Jun Xu</i>	

Lexical semantics and Ontologies

- Exploiting Lexicalized Statistical Patterns in Chinese Linguistic
Analysis 238
Yu Zhao and Maosong Sun

Language Resources and Annotation

- Development of Traditional Mongolian Dependency Treebank..... 247
Xiangdong Su, Guanglai Gao, and Xueliang Yan
- Chinese Sentence Compression: Corpus and Evaluation 257
*Chunliang Zhang, Minghan Hu, Tong Xiao, Xue Jiang,
Lixin Shi, and Jingbo Zhu*

Machine Translation

- Graphic Language Model for Agglutinative Languages: Uyghur as
Study Case..... 268
Miliwan Xuehelaiti, Kai Liu, Wenbin Jiang, and Tuergen Yibulayin
- i*CPE: A Hybrid Data Selection Model for SMT Domain Adaptation ... 280
*Longyue Wang, Derek F. Wong, Lidia S. Chao, Yi Lu, and
Junwen Xing*
- Multi-classifier Combination for Translation Error Detection..... 291
Jinhua Du, Junbo Guo, Sha Wang, and Xiyuan Zhang

Speech Recognition and Synthesis

- Automatic Discrimination of Pronunciations of Chinese Retroflex and
Dental Affricates..... 303
Akemi Hoshino and Akio Yasuda
- A New Word Language Model Evaluation Metric for Character Based
Languages 315
Peilu Wang, Ruihua Sun, Hai Zhao, and Kai Yu

Tagging and Chunking

- Bidirectional Sequence Labeling via Dual Decomposition 325
Zhiguo Wang, Chengqing Zong, and Nianwen Xue
- Semantic Analysis of Chinese Prepositional Phrases for Patent Machine
Translation..... 333
Renfen Hu, Yun Zhu, and Yaohong Jin

Large-scale Knowledge Acquisition and Reasoning

Massive Scientific Paper Mining: Modeling, Design and Implementation	343
<i>Yang Zhou, Shufan Ji, and Ke Xu</i>	
Author Index	353

Improving Chinese Word Segmentation Using Partially Annotated Sentences

Kaixu Zhang¹, Jinsong Su¹, and Changle Zhou²

¹ Xiamen University, Xiamen, Fujian, 361005, China
kareyzzhang@gmail.com, jssu@xmu.edu.cn

² Institute of Artificial Intelligence
Xiamen University, Xiamen, Fujian, 361005, China
dozero@xmu.edu.cn

Abstract. Manually annotating is important for statistical NLP models but time-consuming and labor-intensive. We describe a learning task that can use partially annotated data as the training data. Traditional supervised learning task is a special case of such task. Particularly, we adapt the perceptron algorithm to train Chinese word segmentation models. We mix conventional fully segmented Chinese sentences with partially annotated sentences as the training data. Partially annotated sentences can be automatically generated from the heterogeneous segmented corpora as well as naturally annotated data such as markup language sentences like wikitexts without any additional manual annotating. The experiments show that our method improves the performances of both supervised model and semi-supervised models.

Keywords: naturally annotation, Chinese word segmentation.

1 Introduction

Chinese words in sentences are not explicitly separated by spaces. Chinese word segmentation is the preliminary task to segment Chinese sentences into words in order to do deeper processing such as part-of-speech tagging and parsing.

Like other natural language processing tasks based on statistical machine learning, annotated data is crucial for the performance of word segmentation. But annotated corpora are limited in size and scope due to the time-consuming and labor-intensive annotating.

Besides semi-supervised methods, using heterogeneous segmented corpus to improve word segmentation is therefore researched in order to make use of more annotated data [1,2]. For example, annotated sentence (b) in Figure 1 can be used together with (a) by using annotation adaptation methods, though the two annotations are not totally consistent. But still, fully annotated data is limited.

On the other hand, there are abundant naturally annotated sentences that also contain clues for word segmentation. For example, the sentence (c) in Figure 1 with brackets is from the Chinese Wikipedia. The bracketed phrase “中美关系” used to make a hyperlink to another page is also a valid phrase in sentence (a) and

(b). Comparing to the sentences intentionally segmented by skilled annotators to make training data, partially annotated sentences by general netizens can be obtained easily. Another example is the anchor texts in the HTML files on the Web, which is indeed web-scale.

- (a) CTB: … 保证_中_美_关系_沿着 …
- (b) MSR: … 发展_中美_关系_的 …
- (c) Wikipedia: … 发展[[中美关系]]的 …

Fig. 1. Different annotated sentences containing the same phrase 中美关系 (Sino-US relations) will be treated in a unified way as the training data in this paper

There are mainly two difficulties to use such partially annotated sentences to improve Chinese word segmentation: the learning algorithm needs to be adapted to learn from partial annotations; a relation is needed to bridge the gap between the arbitrary annotation by netizens and the annotation of words.

Motivated by the related work [3], we formally define the learning task using partially annotated data and propose an adapted perceptron algorithm that can learn from partially annotated data for both supervised and semi-supervised learning (Section 2). Fully and partially annotated sentences are mixed and not distinguished in this algorithm.

A span-based representation is used to represent the information of the partially annotated sentences for word segmentation (Section 3). In such representation, sentences annotated with brackets (Figure 1 (c)) and sentences from heterogeneous corpus (Figure 1 (b)) can be treated in the same way.

With a word-based word segmentation model, experiments are conducted on the Chinese Treebank 5 (Section 5). Sentences from the MSR corpus, People’s Daily corpus as well as Baidu Baike (a Chinese wikipedia-like website) are used as partially annotated sentences to improve the performance of the baseline model.

Our contribution is twofold: 1) we proposed an algorithm for word segmentation with partially annotated data which can treat various resources as partially annotated data; 2) we use the naturally annotated sentences provided by common netizens as a resource to improve the performance of word segmentation.

2 Learning with Partially Annotated Data

2.1 Partially Annotated Data as Training Data

The training examples of supervised classification are $\{(x_i, y_i)\}$, where $y_i \in \text{GEN}(x_i)$ and $\text{GEN}(x_i)$ is the set of all possible classes for x_i . For any input x_i , a unique y_i is given as the gold standard output.

For a partially annotated example x_i , the unique gold standard output can not be determined by using only the partial annotation. Instead, a nonempty subset

Inputs: training example $\{(x_i, Y_i)\}$
Initialization: set $\Lambda = 0$
Output: Averaged parameters $\frac{\sum \Lambda_i^t}{TN}$

- 1: **for** $t = 1 \dots T, i = 1 \dots N$
- 2: calculate $z_i = \arg \max_{z \in \text{GEN}(x_i)} \Phi(x_i, z) \cdot \Lambda$
- 3: **if** $z_i \notin Y_i$ **then**
- 4: calculate $y_i = \arg \max_{y \in Y_i} \Phi(x_i, y) \cdot \Lambda$
- 5: $\Lambda = \Lambda - \Phi(x_i, z_i) + \Phi(x_i, y_i)$
- 6: **set** $\Lambda_i^t = \Lambda$

Fig. 2. Averaged perceptron algorithm used for learning from partially annotated data

Y_i of $\text{GEN}(x_i)$ which contains the unknown gold standard output is given. The training examples are thus represented as $\{(x_i, Y_i)\}$, where $\emptyset \subset Y_i \subset \text{GEN}(x_i)$ and $y_i \in Y_i$.

A full annotated example can be seen as a special partially annotated example where $Y_i = \{y_i\}$.

2.2 Perceptron Algorithm for Partially Annotated Data

Collins [4] proposed a perceptron algorithm for structured classification tasks such as part-of-speech tagging. Since it is widely used for Chinese word segmentation, we decide to adapt this algorithm for partially annotated data.

The adapted algorithm is shown in Figure 2 which is similar to the related work [3]. The adapted algorithm is a natural extension of the traditional one [4]. When $Y_i = \{y_i\}$ holds for all the training example, the adapted perceptron algorithm degenerates to the traditional one.

This algorithm can not learn from unannotated sentences. For any unannotated example $(x_i, \text{GEN}(x_i))$, since $z_i \in \text{GEN}(x_i)$ is always true, the updating in the **if** statement will never be executed.

Additionally, we find that the convergence to the expected optimum of this adapted algorithm is not theoretically guaranteed. But fortunately, this algorithm works well in practice as we will show.

2.3 Self-training with Partially Annotated Data

Partially annotated sentences can be also used for semi-supervised algorithms such as self-training.

Figure 3 shows a self-training algorithm which uses partially annotated sentences in the training process. In Step 4, we use the margin to define the confidence:

$$\text{conf}_i = \Phi(x_i, z_i) \cdot \Lambda - \max_{z \neq z_i} \Phi(x_i, z) \cdot \Lambda \quad (1)$$

There are two differences between our algorithm and the conventional self-training algorithm. First, in Step 2, examples that not fully annotated in P are also used to train the model (we call it “p_train” in the experiments). Second,

Inputs:

Fully annotated example set \mathcal{F}
 Partially annotated example set \mathcal{P}

Output:

Model parameter Λ

Algorithm:

- 1: Loop for k -iterations:
- 2: use \mathcal{F} and \mathcal{P} to train parameter Λ'
- 3: use Λ' to segment \mathcal{P}
- 4: move q examples with high
 confidence from \mathcal{P} to \mathcal{F}
- 5: use \mathcal{F} to train parameter Λ

Fig. 3. Self-training algorithm with partially annotated data

in Step 3, when segmenting an example (x_i, Y_i) in \mathcal{P} , the search space of the decoding is limited in the set Y_i (we call it “p-predict” in the experiments).

2.4 Distributed Learning for Large-Scale Training Data

As we mentioned that available partially annotated sentences are large-scale, there are two reasons to use distributed learning for large-scale training data. First, when partially annotated sentences are much more than fully annotated sentences, the learning is harder to converge. Second, our distributed method is faster and is suitable for incremental learning.

Suppose we have n sets of training examples. The parameters of the trained model using these sets are $\Lambda^{(1)} \dots \Lambda^{(n)}$, respectively. Then we calculate the parameters of the final model as:

$$\lambda_i = \frac{\sum_{k=1 \dots n} \lambda_i^{(k)}}{\sum_{k=1 \dots n} \mathbb{1}_{\lambda_i^{(k)} \neq 0}} \quad (2)$$

where $\lambda_i^{(k)}$ is the i -th parameter of the k -th model. Note that the denominator in this equation is not n . The reason is that when $\lambda_i^{(k)} = 0$, it is not because this feature is not important, but because this feature is unseen in the training process of the k -th training data.

For incremental learning, used sentences are not needed to be stored. When new training data is acquired, we only need to train a new model $\Lambda^{(n+1)}$ using the new data and then update the parameter Λ without using any old data.

3 Partially Annotated Sentences for Chinese Word Segmentation

Now we narrow down our discussion to the Chinese word segmentation task.

A raw sentence x is a Chinese sentence where no spaces are presented to separate words, while a segmented sentence is a sentence where words are separated by spaces. For example, “发展中美关系” is a raw sentence, and “_发_展_中_美_关_系_” is one of the possible segmented sentences corresponding to the raw sentence.

We use a span set (a set of spans) as z to represent words in a segmented sentence. The corresponding span set z for the segmented sentence above is $\{\langle 0, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 4, 6 \rangle\}$, the two numbers in $\langle 0, 2 \rangle$ are the indexes of the beginning and end of the word “发展”.

For a training sentence x with partial annotation such as heterogeneous segmented corpus or wikitexts, y can not be determined. Instead, we need to define the set $Y = \{z_j\}$ which include the gold standard output ($y \in Y$) and exclude some impossible outputs. Before giving the definition of Y , we introduce some basis concepts in the next subsection.

3.1 Agreement between Span Sets

We used spans in the span set z to indicate words. Note that spans can also be used to indicate other linguistic components such as morphemes and phrases. Roughly speaking, all these components of a sentence are organized as a hierarchical tree. In other words, following the definition by Klein and Manning [5], the spans of any two components will never cross:

Definition 1. Two spans $\langle b, e \rangle$ and $\langle b', e' \rangle$ (without loss of generality, $b \leq b'$) *disagree* (or *cross*) if and only if $b < b' < e$ and $b' < e < e'$. Otherwise, they *agree* (or are *non-cross*).

For example, span $\langle 2, 5 \rangle$ agree with $\langle 0, 2 \rangle$ and $\langle 2, 4 \rangle$ but disagree with $\langle 0, 3 \rangle$.

Furthermore, we can define the “agree” relation between span sets:

Definition 2. Two span sets S and S' *agree* (denoted as $S \sim S'$) if and only if any span in S agrees with any span in S' .

Then we can give an assumption about the nature of Chinese:

Assumption 0. Span set of words agree with span set of morphemes or phrases in the same sentence.

This assumption may be widely accepted by linguists and is the basis assumption to define partially annotated examples. Although we do not directly use this assumption in this paper, it is the motivation that we choose spans and the “agree” relation to define Y . In the following subsections we will give two more assumptions based on our observation. They may not always hold as the previous one but can be used to define Y effectively.

3.2 Partially Annotated Sentences from Heterogeneous Segmented Corpus

For a sentence x from a heterogeneous segmented corpus, the given gold standard output y' may be different with the gold standard output y we expected. Using (x, y') as the training data will result in bad performance.

We establish a relation between y' and y by given the following assumption:

Assumption 1: Span sets of words in different annotation specifications agree with each other ($y \sim y'$).

This assumption is based on that most of the inconsistency of word definition between different annotation specifications is about the granularity of words. This means that a word under one annotation specification is generally still a word, phrase or morpheme in another annotation specification.

Thus we can define partially annotated examples as

$$(x, Y) := (x, \{z|z \sim y'\}) \quad (3)$$

The set Y is defined by using heterogeneous annotation y' . And with Assumption 1, we have $y \in Y$.

3.3 Partially Annotated Sentences from Wikitexts

Based on our observation, although the annotation in wikitexts is more arbitrary, bracketed texts are still usually words, phrases or morphemes. So similar to the method we used for the heterogeneous corpora, we give an assumption:

Assumption 2: Span set of words and span set of bracketed texts of the same sentence agree with each other.

The partially annotated sentences can thus be defined as

$$(x, Y) := (x, \{z|z \sim b\}) \quad (4)$$

where b is the span set of bracketed texts.

3.4 Mixed Training Data

It is not sufficient that we only use partially annotated sentences defined above as the training data. If so, the algorithm may result in an unexpected optimum that the model segments every single character as a word. Those results will agree with any span sets.

In practice, we mix the fully segmented sentences and the partially annotated sentences and randomly shuffle them as the training data. And our learning algorithm can treat them in the same way.

4 Related Work

In recent years, learning with partially annotated data is concerned by researchers of machine learning [6] as well as natural language processing. Partially annotated data can be used for corpus construction [7], sequence labeling [8], syntactic parsing [9,10] and other NLP tasks [11]. Our algorithm can be seen as a version of the latent structure perceptron [12] which can learn from examples with hidden variables [3]. Zettlemoyer and Collins [13] used similar algorithm for semantic parsing.

We use self-training [14] for our task. Other semi-supervised learning methods such as co-training [15] may also benefit from partially annotated sentences with our method.

Jiang et al. [16] used model trained using heterogeneous segmented corpus to generate new features to improve the performance of joint word segmentation and part-of-speech tagging model. Sun and Wan [2] further used the re-training method to transform the heterogeneous corpus in order to use it directly as the training data. Jiang et al. [1] further used iterative annotation transformation with predict-self reestimation to improve the performance.

New features for word segmentation can also be generated based on the statistical information of the unannotated corpus [17,18,19,20]. Punctuation marks can be seen as artificial annotations for natural language. Li and Sun [21] used the punctuation marks in the unsegmented corpus as clues for word boundaries. Spitzkovsky et al. [22] used hyper-text annotations for unsupervised English parsing.

Our model for word segmentation [23] is mainly motivated by the word-based word segmentation model proposed by Zhang and Clark [24,25] and the linear-time incremental shift-reduce parser proposed by Huang and Sagae [26].

5 Experiments

We use Penn Chinese Treebank 5 (CTB) as the main corpus of our experiments. The partitions of training set (18,086 sentences) and test set are the same with [25]. We use the training data of the MSR corpus from SIGHAN bake-off 2005 (86,924 sentences) and People’s Daily (PD) corpus from Peking University (294,239 sentences) as the heterogeneous corpora.

As the source files of Chinese Wikipedia contain both simplified and traditional Chinese characters and the translation method is not straightforward, we turn to use another wiki-like site Baike from Baidu¹ containing only simplified characters. One million sentences with brackets are used, which is still a small part of the total sentences with brackets that can be extracted.

We use a word-based Chinese word segmentation system [23] which won the first place in the CIPS-SIGHAN bakeoff 2012 [27] as our baseline model². The feature templates are listed in table 1. In all the semi-supervised experiments the parameter k in the self-training algorithm (Figure 3) is set to 10. Since there are no hyper-parameters that are tuned, we directly show the results on the test set instead of the development set.

F-score [28] is used for the evaluation.

5.1 Supervised Learning with Partially Annotated Data

We mix the training data of CTB and the partially annotated sentences generated from other resources together as the training data for supervised learning.

¹ <http://baike.baidu.com/>

² <https://github.com/zhangkaixu/isan>

Table 1. Feature templates

action-based	$\langle \mathbf{a01}, a_{i-2}, a_{i-1} \rangle$
character-based	$\langle \mathbf{c01}, c_{i-2}, a_{i-1} \rangle, \langle \mathbf{c02}, c_{i-1}, a_{i-1} \rangle, \langle \mathbf{c03}, c_i, a_{i-1} \rangle$ $\langle \mathbf{c04}, c_{i-3}, c_{i-2}, a_{i-1} \rangle, \langle \mathbf{c05}, c_{i-2}, c_{i-1}, a_{i-1} \rangle,$ $\langle \mathbf{c06}, c_{i-1}, c_i, a_{i-1} \rangle, \langle \mathbf{c07}, c_i, c_{i+1}, a_{i-1} \rangle$
word-based	$\langle \mathbf{w01}, \mathbf{w}_0 \rangle, \langle \mathbf{w02}, \mathbf{w}_0 \rangle$ $\langle \mathbf{w03}, \mathbf{w}_0 , \mathbf{w}_0[0] \rangle, \langle \mathbf{w04}, \mathbf{w}_0 , \mathbf{w}_0[-1] \rangle, \langle \mathbf{w05}, \mathbf{w}_0[0], \mathbf{w}_0[-1] \rangle$ $\langle \mathbf{w06}, \mathbf{w}_{-1}[-1], \mathbf{w}_0[-1] \rangle, \langle \mathbf{w07}, \mathbf{w}_{-1} , \mathbf{w}_0 \rangle, \langle \mathbf{w08}, \mathbf{w}_{-1}, \mathbf{w}_0 \rangle$ $\langle \mathbf{w09}, \mathbf{w}_0[0], c_i \rangle, \langle \mathbf{w10}, \mathbf{w}_0[-1], c_i \rangle$

Table 2. Results of supervised learning with partially annotated sentences

Training Set	F1
CTB	0.9745
CTB + MSR _{partial}	0.9767
CTB + $\frac{1}{4}$ PD _{partial}	0.9773
CTB + PD _{partial}	0.9752
CTB + Baike_50K _{partial}	0.9761

Table 2 shows the results. Partially annotated sentences generated from both heterogeneous corpus and wikipedias can improve the performance. Note that the PD corpus is much larger than the MSR corpus. Probably because the converging is harder when the rate of partially annotated sentence is high, we find that using a quarter of these sentences are even better than using all at once.

5.2 Self-training with Partially Annotated Data

Three different self-training algorithms are performed and compared. The conventional self-training algorithm without using any partially annotated information is denoted as “baseline”. The self-training algorithm proposed by us in Figure 4 is denoted as “p_train+p_predict”. We also use an algorithm like “p_train+p_predict” but does not use partially annotated sentences in the training process which is denoted as “p_predict”.

Experiment results are shown in Figure 4. All these three self-training algorithms outperform the supervised algorithm. The algorithm “p_train+p_predict” can always improve the performance. For a corpus like MSR where the partial annotation is relatively rich, only using “p_predict” can also improve the performance. But for Baike_50K where annotated information is rare, the difference between the performances of “p_predict” and “baseline” is not obvious.

5.3 Distributed Learning with Large Data

From the experiment results of the supervised learning we already find that simply using large partially annotated dataset is not always helpful.

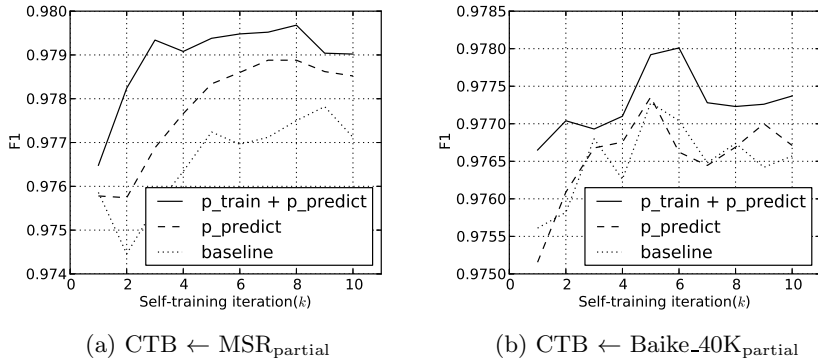


Fig. 4. Results of the self-training algorithm

In this subsection we first divide the PD corpus into four parts and use the distributed learning to train the models.

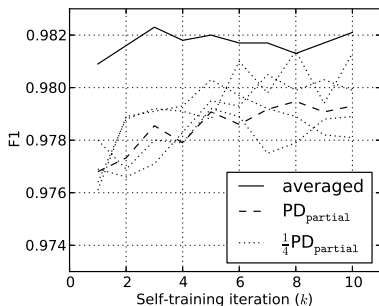


Fig. 5. Results of self-training using PD corpus. Distributed learning outperforms the baseline.

Figure 5 shows the results. The dotted line is the curve of using the whole PD corpus in the self-training algorithm. Slashed lines are four curves of using a quarter of the PD corpus, respectively. The solid line is the curve of the averaged model based on these four small models using Equation 2.

Finally, we perform the distributed self-training with one million sentences from Baike (divided into 25 sets). Table 3 shows the final results of our method and the results of related work. It is not surprise that our method do not outperform the annotation adaptation method [1], since we only treat the heterogeneous corpus as a partial annotated corpus. But with the same method, we can make use of the partial annotation information in the wiktexsts. Our word segmentation model using one million Baike sentences is comparative to the joint word segmentation and part-of-speech tagging model [20] using approximately 208 million additional words from Xinhua newswire.

Table 3. Final results of our method are compared with related work

Training Set	F1
CTB	0.9745
CTB + PD _{partial}	0.9821
CTB + Baike_1M _{partial}	0.9810
CTB + PD [16]	0.9815
CTB + PD [1]	0.9843
CTB + Gigaword [20]	0.9811

6 Discussion and Conclusion

We presented a learning method with partially annotated sentences for Chinese word segmentation. Naturally annotated data such as wikipedias can be treated as partially annotated sentences and be used as training data together with fully annotated sentences. Our method is potentially suitable for domain adaptation where in-domain fully segmented sentences are limited.

Our work is just a primary study that uses the partial annotation information for Chinese language processing. In the future, we will try to use similar method for word-based active learning and syntax parsing.

Acknowledgments. The authors want to thank ZHANG Junsong from the cognitive lab and SHI Xiaodong and CHEN Yidong from the NLP lab of Xiamen University for the support of experiments.

The authors are supported by NSFC (Grant No. 61273338), Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20120121120046) and Natural Science Foundation of Fujian Province (Grant No. 2010J01351).

References

1. Jiang, W., Meng, F., Liu, Q., Lü, Y.: Iterative annotation transformation with predict-self reestimation for Chinese word segmentation. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, pp. 412–420. Association for Computational Linguistics (July 2012)
2. Sun, W., Wan, X.: Reducing approximation and estimation errors for Chinese lexical processing with heterogeneous annotations. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Long Papers, Jeju Island, Korea, vol. 1, pp. 232–241. Association for Computational Linguistics (July 2012)
3. Fernandes, E., dos Santos, C., Milidiú, R.: Latent structure perceptron with feature induction for unrestricted coreference resolution. In: Joint Conference on EMNLP and CoNLL - Shared Task, Jeju Island, Korea, pp. 41–48. Association for Computational Linguistics (July 2012)

4. Collins, M.: Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms, pp. 1–8 (2002)
5. Klein, D., Manning, C.D.: A generative constituent-context model for improved grammar induction. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp. 128–135. Association for Computational Linguistics (July 2002)
6. Lou, X., Hamprecht, F.: Structured learning from partial annotations. arXiv:1206.6421 (June 2012)
7. Neubig, G., Mori, S.: Word-based partial annotation for efficient corpus construction. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010), Valletta, Malta. European Language Resources Association, ELRA (2010)
8. Tsuboi, Y., Kashima, H., Mori, S., Oda, H., Matsumoto, Y.: Training conditional random fields using incomplete annotations. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Manchester, UK, pp. 897–904. Coling 2008 Organizing Committee (August 2008)
9. Mirroshandel, S.A., Nasr, A.: Active learning for dependency parsing using partially annotated sentences. In: Proceedings of the 12th International Conference on Parsing Technologies, Dublin, Ireland, pp. 140–149. Association for Computational Linguistics (October 2011)
10. Flannery, D., Miyao, Y., Neubig, G., Mori, S.: Training dependency parsers from partially annotated corpora. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, pp. 776–784. Asian Federation of Natural Language Processing (November 2011)
11. Fernandes, E.R., Brefeld, U.: Learning from partially annotated sequences. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part I. LNCS, vol. 6911, pp. 407–422. Springer, Heidelberg (2011)
12. Yu, C.N.J., Joachims, T.: Learning structural SVMs with latent variables. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 1169–1176. ACM, New York (2009)
13. Zettlemoyer, L., Collins, M.: Online learning of relaxed CCG grammars for parsing to logical form. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, pp. 678–687. Association for Computational Linguistics (June 2007)
14. McClosky, D., Charniak, E., Johnson, M.: Effective self-training for parsing. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, New York City, USA, pp. 152–159. Association for Computational Linguistics (June 2006)
15. Sarkar, A.: Applying co-training methods to statistical parsing. In: Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies, NAACL 2001, Stroudsburg, PA, pp. 1–8. Association for Computational Linguistics (2001)
16. Jiang, W., Huang, L., Liu, Q.: Automatic adaptation of annotation standards: Chinese word segmentation and POS tagging - a case study. In: Proceedings of the 47th ACL, Suntec, Singapore, pp. 522–530. Association for Computational Linguistics (August 2009)
17. Feng, H., Chen, K., Deng, X., Zheng, W.: Accessor variety criteria for Chinese word extraction. *Computational Linguistics* 30(1), 75–93 (2004)

18. Zhao, H., Kit, C.: Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In: The Sixth SIGHAN Workshop on Chinese Language Processing, pp. 106–111 (2008)
19. Sun, W., Xu, J.: Enhancing Chinese word segmentation using unlabeled data. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK, pp. 970–979. Association for Computational Linguistics (July 2011)
20. Wang, Y., Kazama, J., Tsuruoka, Y., Chen, W., Zhang, Y., Torisawa, K.: Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, pp. 309–317. Asian Federation of Natural Language Processing (November 2011)
21. Li, Z., Sun, M.: Punctuation as implicit annotations for Chinese word segmentation. *Computational Linguistics* 35(4), 505–512 (2009)
22. Spitzkovsky, V.I., Jurafsky, D., Alshawi, H.: Profiting from mark-up: Hyper-text annotations for guided parsing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 1278–1287. Association for Computational Linguistics (July 2010)
23. Zhang, K., Sun, M., Zhou, C.: Word segmentation on Chinese micro-blog data with a linear-time incremental model. In: Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, Tianjin, China, pp. 41–46. Association for Computational Linguistics (December 2012)
24. Zhang, Y., Clark, S.: Chinese segmentation with a word-based perceptron algorithm, Prague, Czech Republic, pp. 840–847. Association for Computational Linguistics (June 2007)
25. Zhang, Y., Clark, S.: Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics (Early Access)*, 1–47 (2011)
26. Huang, L., Sagae, K.: Dynamic programming for linear-time incremental parsing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 1077–1086. Association for Computational Linguistics (July 2010)
27. Duan, H., Sui, Z., Tian, Y., Li, W.: The cips-sighan clp 2012 Chinese word segmentation on microblog corpora bakeoff. In: Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language Processing, Tianjin, China, pp. 35–40. Association for Computational Linguistics (December 2012)
28. Emerson, T.: The second international Chinese word segmentation bakeoff. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea, pp. 123–133 (2005)

Chinese Natural Chunk Research Based on Natural Annotations in Massive Scale Corpora*

Exploring Work on Natural Chunk Recognition Using Explicit Boundary Indicators

Zhi-e Huang, En-dong Xun, Gao-qi Rao, and Dong Yu

BLCU, International R&D Center for Chinese Education
{hze_blcu, raogaoqi_fj}@163.com, {edxun, yudong_blcu}@126.com

Abstract. Great changes in Natural Language Processing (NLP) research appear with the rapid inflation of corpora scale. NLP based on massive scale natural annotations has become a new research hotspot. We summarized the state of art in NLP based on massive scale natural annotated resource, and proposed a new concept of “Natural Chunk”. In the paper, we analyzed its concept and properties, and conducted experiments on natural chunk recognition, which exhibit the feasibility of natural chunk recognition based on natural annotations. Chinese natural chunk research, as a new research direction in language boundary recognition, has positive influences in Chinese computing and promising future.

Keywords: Massive scale corpora, Natural Annotation, Natural Chunk, Natural Chunk recognition.

1 Introduction

Language boundary plays a significant role in human language acquisition. Language boundary is a basic explicit feature of language unit, on which research is known as a basic research in linguistic [1]. In Chinese, different terms are used in description of the concept of “word”, like “syllable word”, “separable word”, “short phrase”, “rhythm word”, “rhythm phrase” etc [2]. Different inner sentence units are defined for various applications in Natural Language Processing (NLP), like Chinese word segmentation, chunking, rhythm phrase recognition etc, which also require different computing methods.

With the development of Internet, as to corpora construction, data scale is no longer the main difficulty. With the improvement of hardware performance and distributed computing technology, algorithm barriers decrease a lot. All these influence the NLP research and attract considerable attentions to exploring methods of automatically acquiring linguistic knowledge from raw corpus. [Sun, 2011] proposed the concept of “NLP based on massive scale natural annotated corpora” and stated that Natural

* Supported by NFSC(61170162), State Language Commission (YB125-42), National Science-technology Support Plan Projects (2012BAH16F00) and the Fundamental Research Funds for the Central Universities(13YCX192).

Language Processing Based on Huge-scale Naturally Annotated Corpora should be the future direction of NLP research [3].

A natural sentence is made up of words, among which exist cohesion and transition relations. In massive scale of corpora, these relations could be relieved by natural annotations. Theory of Prefabricated Chunk, rooted in cognitive linguistic, states the existence of lots of directly used poly-word strings with solidification and prefabrication. Combining the theory of Prefabricated Chunk with Natural Annotation oriented NLP research, we put forward a new language unit, define the solidified strings that frequent and stably appear in various contexts as ‘Natural Chunks’.

Natural chunk is a kind of language unit observed from pragmatic level. Natural chunk recognition aims to meet various application requirements on language boundaries within a united framework. Boundaries’ strength should be maintained, so as to support the further recognition of other language units’ boundaries among a sentence. The task of natural chunk recognition is in fact the word boundary prediction based natural annotation in massive scale of corpora. This task contains 3 parts, namely natural annotation mining, natural chunk boundary modeling, and evaluation. By parameters modification, different chunking for one sentence could be approached. Existed relative work shows that chunking based on natural annotation is feasible, and has positive influences in Chinese computing.

2 Language Computing Based on Natural Annotated Boundary Knowledge

Natural annotation resource consists of various resource data generated by users in all means, like web pages, forums, twitters, Wikipedia, etc. In the view of natural language processing (NLP), these data can be seen as partially tagged. And the natural annotations are quite worthy to be utilized.

On its application in NLP, first, massive scale is the necessary condition to make Natural Annotation computing into play. Second, shallow processing is the basic tone of calculation on massive scale corpora. At last, professional linguist knowledge should not be refused in any case in analysis and processing [3]. Relative researches in Chinese word segmentation, rhythm phrase recognition, information extraction, social computing, sentiment analysis, text classification and information retrieval have come to appear in recent years.

Natural annotations are rich of lexical information. Experimental and research results indicate that they can benefit language boundary recognition. Language boundary recognition researches took advantages of natural annotations are mainly about Chinese word segmentation and rhythm phrase recognition. [Rao etc.,2013] explored the lexicon knowledge containing by punctuations, Latin letters and Arabic number in small data set [4]. [Zhongguo Li etc., 2009] used punctuation as segmentation features offering positive cases, increasing word segmentation performance of CRF model, especially for OOV (words Out Of Vocabulary) in BakeOff test [5]. [Yuhang Yang etc., 2008] research about Chinese term extraction based on delimiter [6]. [Xing Li etc., 2006] introduced punctuations as a important feature into hierarchical Chinese chunking [7].

[Thomas etc., 2005] increased precision on Chinese-English sentence pair alignment by adding punctuation as feature [8]. [Qian Yili etc., 2008, Xun Endong etc., 2005] proposed method of building rhyme structure binary tree, based on the feature of punctuation location. High precision was reached with low training cost [9,10]. [Valentin etc., 2010] increased unsupervised chunking 5 percent with features of HTML tags (anchor, bold, italic and underline). Punctuations were also used in chunking [11]. [Valentin etc., 2011; Weiwei Sun etc., 2011] merged punctuations into other traditional statistic features, resulting to better performance in OOV [12,13].

In brief, Natural annotations are unconsciously annotated by users, which avoid the problem of tagging cost. With massive scale corpora, both explicit and implicit annotation mining will partially conquer the difficulty in quality assurance and quantity limit, benefitting various tasks in NLP.

3 Chinese “Natural Chunk”

Theory of Prefabricated Chunk (PC), rooted in cognitive linguistic, states the existence of lots of directly used poly-word strings with solidification and prefabrication. Prefabrication indicates that syntax generation and inner syntax analysis are not in need in the usage of PC each time. A natural sentence is made up of words, among which exist hierarchy and cohesion relations. With a massive scale of corpora, these properties could be relieved by the natural annotations which richly contain boundary information, appear in a specific form of language unit. This language unit is defined as “Natural Chunk”.

Definition. The language units that continuous stable and frequent appear with distinctive boundary features in massive scale corpus are ‘Natural Chunks’.

In Chinese, natural annotations like punctuations, Arabic numbers, Latin letters match the definition of natural chunk, and they are special ones.

Properties of “Natural Chunks”

- (a) Integrity (inner cohesion). A natural chunk is a continuous string. Conventionalized usages are often seen as natural chunks like “与此同时(at the same time)”, “也就是说(that is to say)”
- (b) Stability. In massive scale corpora, frequently used in various contexts.
- (c) Boundary Features. Natural chunks is strings with distinct boundaries. The boundary information offered by natural annotations in massive scale corpora. Different from other language units which are bounded with syntactical rules.
- (d) Application Oriented. The properties of natural chunks depend on the properties of the natural annotations used in the recognition progress. For various applications, the mining and usage of various natural annotations adopting in the process varied, different but rational natural chunk recognition can be obtained. Moreover, their evaluations should also adapt to the specific application.

Properties of “integrity” and “stability” are quite similar to the principles of Chinese word segmentation. If punctuations are used as Natural Annotation in recognition, chunks also often match rhyme structures in one sentence. Natural Chunking has advantages in Chinese language boundary computation, for it is not bound with syntactical rules as the others do, and also due to its unsupervised mining process in massive scale corpora.

4 Natural Chunk Recognition in Massive Scale Corpora

The work in this section is consisted by three parts, namely the mining of natural annotations with distinctive boundary information, the boundary modeling and evaluation. All these work are concentrating on the boundary computing of natural chunks.

4.1 Natural Annotations with Distinctive Boundary Information

We use BIC (Boundary Information Carrier) to signify natural annotations carrying distinctive boundary information expressed. Base on whether a BIC is intuitive and easy to extract, classify BICs into explicit BICs and implicit BICs. A explicit BIC should be intuitive and easy to extract. To be specific, Punctuation, line break, Arabic numbers and Latin letters are explicit BICs, because they are easy to be identified and extracted from Chinese sentences, since they do not belong to Chinese character set and never associate with other Chinese characters as a word. Ex1 present how punctuations and Arabic numbers separate the chunk out of a sentence. “的” (of), “与” (and) and “对垒” (confront) are implicit BICs. By the way, using explicit BICs to acquire implicit BICs, is one of our studies in the future.

Ex.1

- (a) …… 。 改革开放以来 ，老百姓开矿治穷……
- (b) “ 站住 ！” 值班的战士大吼一声……
- (c) ……截止收盘沪指报2293.08点，涨4.55点……
- (d) 在市场上站住脚 与 站稳脚 的 思路同样有差异。
- (e) 她们将同老对手、实力强劲的[org] 对垒 ……

As subset of natural annotations, BICs contain rich linguistic knowledge, including shallow formats and pragmatic patterns etc. High coverage of various lingua phenomena could be approached. By iterations, more BICs especially implicit ones could be gained. Ex.3 shows how to gain a chunk “篡改财务账目” (tampering with financial accounts) in d) from “改革开放” (reform and opening up) in a) by “特别是”(especially) in b) and “通过采取” (by taking) in c). BIC mining, boundary strength computing and iteration strategy is critical in this part.

Ex.2

- (a) 罗干说，改革开放以来，中国旅游业……
- (b)。特别是←改革开放以来，人民子弟兵……
- (c) 特别是→通过采取“切割”、“站票”……
- (d)。通过采取→篡改财务账目的方式贪污赃款

4.2 Natural Chunk Boundary Modeling

Natural chunks are strings with distinctive boundaries. It is crucial to utilize context information in boundary modeling. On the other hand, a natural chunk is stably and frequently used in massive scale corpora. Its inner cohesion is another influencing factor of boundary modeling. A natural chunk's boundary could be gained at the low point of inner cohesion. In brief, both inner cohesion and autonomy in context are import features in boundary modeling.

Similar with Chinese word segmentation, frequency, mutual information (MI), boundary entropy (BE) and accessor variety (AI) are often used to describe the cohesion and isolation of a “word” [14]. [Hanshi Wang etc, 2011] proposed algorithm of “Evaluation-Segmentation-Adaption” merging boundary entropy, frequency and length, and obtained a segmentation after a convergence iteration process [15].

Natural chunks are flexible in granularity. However within a specific application, a sentence has only one specific and rational natural chunk recognition result, matching the needs of various applications by parameter adaption. Boundary strength is necessary information in natural chunk recognition. Any strings in a sentence, its isolation to be a natural chunk could also be described by boundary strength in boundary modeling. We will investigate the cohesion and isolation of natural chunk and its boundary modeling combining both.

4.3 Evaluation of Natural Chunk Recognition

Natural chunk recognition, like Chinese word segmentation as well as rhythm recognition, is a kind of research on language boundary identification. But it should be pointed out that natural chunk recognition aims to gain a united framework for various kinds of language boundary recognition. Within the framework, by modification of parameters and decoding strategies, different needs of applications in language boundaries should be matched. And further evaluations should also fit the application needs.

To make it simple, Chinese lexicon is in natural small natural chunks solidified in pragmatics, while rhythm phrase can be viewed as natural chunks in coarse granularity. From this point, fine natural chunking of small granularity for segmentation while coarse granularity for phrase recognition.

If Chinese word segmentation is viewed as application context, parameter tuning can be realized comparing with tagged corpus, and made its chunking result close to the granularity of word. Its test result would also be evaluated with the standard of word segmentation. Similarly is the application context of rhyme phrase.

5 Experiments

5.1 Corpus and Dataset

In this paper, the corpus in use is a massive scale balanced monolingual corpus, consisted of Beijing Language and Cultural University International R&D Centre for Chinese Education. The corpus contains News (People Daily), Literature, Weibo (Chinese Tweeter, 3 months data) and Blog (3 months data), of total size 102.8 GB (ASCII). Size of each part is shown in Table.1.

Table 1. Data Distribution of each Part of Corpus

Type	News	Literature	Weibo	Blog	Total
Scale (B, ASCII)	6.26G	8.29G	35.0G	53.3G	102.8G

5.2 Lexicon Knowledge in BICs

Punctuations are special natural chunks. In Chinese, punctuation is a close set and easy to recognize, carrying distinctive boundary information. All punctuations in the previously mentioned massive scale corpus are replaced by “ Δ ”, and used as a segmentations’ symbol and segmented the corpus.

The Modern Chinese Dictionary (5th version), ‘MCD5’ for short, contains 62777 different Chinese word types. Substrings of 1-3 characters and 4-8 characters are extracted from the segmented corpus mentioned above. Among them, over 96% words of MCD5 have been covered. In fact, [Rao Gaoqi etc, 2013] reported in the small data set of 350MB (ASCII), 87.84% words can be covered. Fig.1 shows the increasing of word coverage with pruning frequency⁴. By then, we can come to an conclusion that explicit Natural annotations like punctuation in big data set richly contain language boundary knowledge.

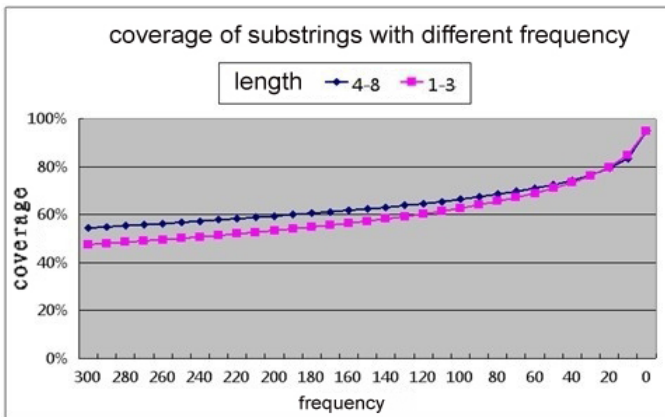


Fig. 1. Words Coverage in Modern Chinese Dictionary (5th version)

5.3 Word Segmentation Experiments with explicit BICs

A string $C_1C_2\cdots C_{i-1}C_iC_{i+1}\cdots C_{n-1}C_n$ can be equivalent to an interval form as $C_1I_1C_2I_2\cdots C_{i-1}I_{i-1}C_iI_iC_{i+1}I_{i+1}\cdots C_{n-1}I_{n-1}C_nI_n$. In interval form, $I_i=1$, when it is a word boundary (B for short) or $I_i=0$. Straightforwardly $p(B_i)=p(I_i=1)$ [16,17].

Assumption 1. Boundary strength (statistically indicates the probability for a character boundary being a word boundary) is only relative to the adjacent context. Therefore in massive corpus, the boundary information carried in BICs could highly cover various language boundaries.

“ Δ ” stands for punctuation. We present $p(B_i)$ in tri-gram form. And we have:

$$\begin{aligned} p(B_i) &= p(C_{i-1}C_iI_i = 1C_{i+1}C_{i+2}) \\ &= p(C_{i-1}C_i, I_i = 1) \times p(C_i, I_i = 1, C_{i+1}) \times p(I_i = 1, C_{i+1}C_{i+2}) \\ &\approx p(C_{i-1}C_i \Delta) \times p((C_i \Delta C_{i+1})) \times p(\Delta C_{i+1}C_{i+2}) \end{aligned} \quad (1)$$

Assumption 2. Boundary strength is positive correlated to the isolation of its context, while negative correlated to the cohesion of its context.

According to the assumption that BICs richly contains boundary knowledge, considering positive correlation exists between boundary strength and its co-occurrence frequency with the punctuations, while negative correlation exists between boundary strength and its co-occurrence frequency of context in a massive scale corpus. Hence having $f(B_i)$ formulated as (4) in a context window of width 4 characters, within which “ Δ ” stands for any punctuation.

$$f(B_i) = \frac{C(C_{i-1}C_i \Delta) \times C(\Delta C_{i+1}C_{i+2})}{C(C_{i-1}C_iC_{i+1}C_{i+2})} \quad (2)$$

Test set1 and 2 are extracted from People’s Daily (Jan. 1998). 5328 sentences from January (19.1 characters in average length) and 5328 sentences from December (19.3 characters in average length).

Table 2. Test Sets

Test Set	Character Number	Character Type	Word Number	Average word Length
Set1	67896	2392	39337	1.7260086
Set2	68424	2261	39803	1.7190664

Tuning point distinction is chosen as decoding strategy. For each character boundary I_i , comparing its previous boundary I_{i-1} and following boundary I_{i+1} , if $p(I_i) \geq p(I_{i-1})$ (or, $f(I_i) \geq f(I_{i-1})$) and $p(I_i) \geq p(I_{i+1})$ (or, $f(I_i) \geq f(I_{i+1})$), then $I_i=1$ and I_i is a word boundary B_i ; or $I_i=0$ and I_i is not a word boundary.

Precision (P in short), Recall (R in short) and F-0.5 were used for evaluation. F-0.5 value combines precision and recall, and it emphasizes precision. They are defined as

formula (3), (4) and (5). A denotes word boundaries by manual annotated, while N_A is set A 's count. B denotes word boundaries tagged by our algorithm, and N_B is the count of set B .

Why not F1? For the natural chunk recognition's result – natural chunks might be somehow rough than the words. As a string “AB” is stably and frequently appear with BICs in a massive scale corpora, according to definition of natural chunk, we might take “AB” as a natural chunk, however in segmentation, it might be take in the form of “A” and “B”. Since we are more interested in natural chunk recognition than rarely Chinese word segmentation, we'd prefer to take F-0.5 value than F-1.

$$P = \frac{A \cap B}{N_A} \times 100\%, \quad (3)$$

$$R = \frac{A \cap B}{N_B} \times 100\% \quad , \quad (4)$$

$$F(0.5) = \frac{(1+0.5^2) \cdot P \cdot R}{0.5^2 \cdot P + R} \times 100\% \quad (5)$$

BE, AV and MI are often used statistic features. We build a baseline system, based on character entropy. Means of left entropy and right entropy are used as description of boundary strength.

Table 3. Performance of Baseline System

Test Set	Type	Avg Len	P	R	F-1	F-0.5
Set1	Blog	2.5988	75.64%	50.24%	60.37%	68.69%
	Literature	2.6040	67.26%	44.58%	53.62%	61.05%
	News	2.5663	76.80%	51.65%	61.76%	69.98%
	Weibo	2.5376	64.91%	44.15%	52.55%	59.33%

Set2	Blog	2.5995	78.39%	51.84%	62.41%	71.11%
	Literature	2.6177	69.77%	45.82%	55.32%	63.17%
	News	2.5626	79.06%	53.04%	63.49%	72.00%
	Weibo	2.5766	66.18%	44.15%	52.97%	60.17%

5.4 Results and Analysis

Table.4 presents the result based on assumption 1, and Table.5 shows the result of experiments based on assumption 2. it is easy to observe that just by the boundary information contained in explicit BICs, tri-gram method could beat baseline system (nearly 18 percentages increased in both precision and F-0.5). Even if we alternate the corpus resource to Sina Blog (different type of writing to People's Daily), this

advantage changes little. By about 3 percentages in precision and 1 percentage in F-0.5 increases.

Mentioned results suggest the effectiveness of the natural boundary computing in massive scale of corpus. It is also easy to observe that stylistic difference influences little on boundary recognize, that will benefit cross domain research.

Table 4. Results of Tri-gram ans Assumption 1 Model

Test Set	Type	Avg Len	P	R	F-1	F-0.5
Set1	Blog	2.197566	93.42%	73.38%	82.20%	88.58%
	News	2.205848	93.34%	73.03%	81.95%	88.42%
Set2	Blog	2.190479	93.57%	73.44%	82.29%	88.71%
	News	2.203955	93.57%	72.99%	82.01%	88.58%

Table 5. Results of Assumption 2 Model

Test Set	Type	Avg Len	P	R	F-1	F-0.5
Set1	Blog	2.4142	96.59%	69.06%	80.54%	89.46%
	Literature	2.4363	95.40%	67.58%	79.12%	88.14%
	News	2.4137	96.57%	69.05%	80.53%	89.44%
	Weibo	2.4352	95.98%	68.03%	79.62%	88.69%
	Mixed (all)	2.4202	96.51%	68.83%	80.35%	89.32%
Set2	Blog	2.4051	96.68%	69.11%	80.60%	89.54%
	Literature	2.4366	95.87%	67.64%	79.31%	88.48%
	News	2.4025	96.57%	69.10%	80.56%	89.46%
	Weibo	2.4199	96.01%	68.20%	79.75%	88.77%
	Mixed (all)	2.4104	96.67%	68.94%	80.48%	89.47%

If we valued 1 for the word boundary in golden standards and 0 for none word boundary. Boundary scoring is normalized by the division of the scoring of the whole sentence. Fig.2 presents the boundary scoring of sentence “迈向充满希望的新世纪” (Towards a new century which is full of hope) and its manual segmentation. An optimal segment point could be found, and sentence will be split into two. Recursively processed by the same steps until the segmented strings match the expectation as a “word”. Fig.3 is the binary segment tree formed in recursive process on the sentence. We could find strong isomorphism of our result with word segmentation standard, and in some levels of binary segment tree, chunk boundaries can also match the rhyme structure of this sentence.

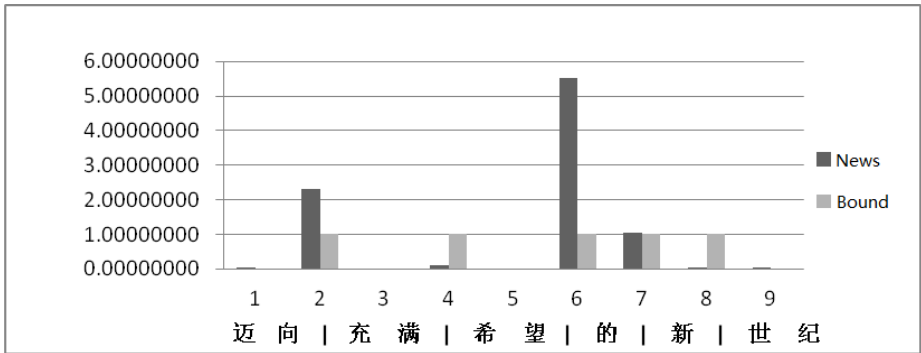


Fig. 2. Boundary Segmentation and Manual Segmentation

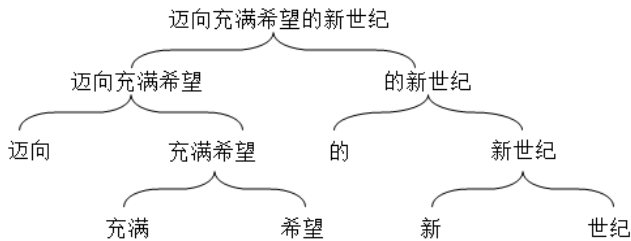


Fig. 3. Binary Segment Tree

Ex.3 indicates the good performance in named entity recognition, especially in long sentences, using punctuations and implicit BICs. “与” (and), “为” (for), “和” (and), “的” (of) etc., are characters (single character word) with strong boundary strength. They are quite worthy to be used in boundary computing. As for future work, it would be natural for researchers to enhance the implicit Natural Annotation utilization as well as modeling both cohesion and autonomy of a string.

Ex.3 Segmentation Examples

- (a) 江泽民 李鹏 乔石 朱镕基 李瑞环 刘华清 尉健行 李岚清 与万名 首都 各界 群众 和劳动 模范 代表 一起 辞旧迎新
- (b) 本报讯 广东鹤山市 直达 香港 九龙的 豪华客车专线 日前开 通
- (c) 恭城瑶族自治县 提供了 成功 的经验
- (d) 刚刚在 英国 首都 伦敦 为争取 英国 政府 释放 智利 前总统 皮诺切特 进行了 三天 游说 活动的 智利 外长 因苏尔萨 将于 n日 赶赴 西班牙 首都 马德里

6 Summary and Future Work

A natural sentence is made up of words, among which exist cohesion and transition relations. In massive scale of corpora, these relations could be relieved by utilizing the co-occurrence with natural annotations in massive scale corpora. Theory of Prefabricated Chunk (PC), rooted in cognitive linguistic, states the existence of lots of directly used poly-word strings with solidification and prefabrication. Combining this phenomena and natural annotation oriented NLP research, we define that the “Natural Chunk” is the solidified, frequent string that stably collocate with various contexts.

Natural chunks are language units observed from pragmatic level. Natural chunk recognition aims to meet various application requirements on language boundaries in a united framework. Information of boundary strength should be contained, so that the further description on different boundaries from characters to sentence could be possible. The task of Natural Chunk recognition is in fact the word boundary prediction based natural annotation in massive scale of corpora. This task contains 3 parts, namely natural annotation mining, chunk boundary modeling, and chunking evaluation. By parameters modification, different chunking for one sentence could be approached. Existed work shows that chunking based on natural annotation is quite effective and has promising future.

As for boundary prediction based on massive scale of corpora, it is a new task serving other relative applications. It is worth noting that, current work still ongoing. Boundary modeling combining isolation and inner cohesion, features selection and pruning criteria are all worthy task waiting to research.

References

1. Liu, C.: Structure and Boundary - A Cognitive Study on Linguistic Expressions. Shanghai Foreign Language Education Press (December 2008)
2. Feng, S.: The multidimensional properties of “word” in Chinese. *Contemporary Linguistics* 3(3), 161–174 (2001)
3. Sun, M.: Natural Language Processing Based on Naturally Annotated Web Resources. *Journal of Chinese Information Processing* 25(6), 26–32 (2011)
4. Rao, G., Xun, E.: Word Boundary and Chinese Word Segmentation. *Journal of Beijing University (Natural Science Edition)* 49(1) (2013)
5. Li, Z., Sun, M.: Punctuation as implicit annotations for Chinese word segmentation. *Computational Linguistics* 35(4), 505–512 (2009)
6. Yang, Y., Lu, Q., Zhao, T.: Chinese Term Extraction Based on Delimiters. In: Conference: Language Resources and Evaluation – LREC (2008)
7. Li, X., Zong, C.: A Hierarchical Parsing Approach with Punctuation Processing for Long Chinese Sentences. *Journal of Chinese Information Processing* 20(4), 8–15 (2006)
8. Chuang, T.C., Yeh, K.C.: Aligning Parallel Bilingual Corpora Statistically with Punctuation Criteria. *Computational Linguistics and Chinese Language Processing* 10(1), 95–122 (2005)
9. Qian, Y.-L., Xun, E.-D.: Prediction of Speech Pauses Based on Punctuation Information and Statistical Language Model. *PR&AI* 21(4), 541–545 (2008)

10. Xun, E.-D., Qian, Y.-L., Guo, Q., Song, R.: Using Binary Tree as Pruning Strategy to identify Rhythm Phrase Breaks. *Journal of Chinese Information Processing* 20(3), 23–28 (2006)
11. Spitzkovsky, V.I., Jurafsky, D.: Profiting from mark-up: Hypertext annotations for guided parsing. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1278–1287 (2010)
12. Spitzkovsky, V.I., Alshawi, H., Jurafsky, D.: Punctuation: Making a Point in Unsupervised Dependency Parsing. In: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pp. 19–28 (2011)
13. Sun, W., Xu, J.: Enhancing Chinese Word Segmentation Using Unlabeled Data. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 970–979 (2011)
14. Zhao, H., Kit, C.: An Empirical Comparison of Goodness Measures for Unsupervised Chinese Word Segmentation with a Unified Framework. In: *International Joint Conference on Natural Language Processing – IJCNLP 2008* (2008)
15. Wang, H., Zhu, J., Tang, S., Fan, X.: A New Unsupervised Approach to Word Segmentation. *ACL* 37(3), 421–454 (2011)
16. Huan, C.-R., Šimon, P., Hsieh, S.-K., Prévot, L.: Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Wordbreak Identification. In: *Proceedings of the ACL 2007 Demo and Poster Sessions*, pp. 69–72 (2007)
17. Li, S., Huang, C.-R.: Chinese Word Segmentation Based on Word Boundary Decision. *Journal of Chinese Information Processing* 24(1), 3–7 (2010)

A Kalman Filter Based Human-Computer Interactive Word Segmentation System for Ancient Chinese Texts

Tongfei Chen¹, Weimeng Zhu¹, Xueqiang Lv³, and Junfeng Hu^{2,*}

¹ School of Electronics Engineering & Computer Science,
Peking University, Beijing, 100871, P.R. China

² Key Laboratory of Computational Linguistics, Ministry of Education,
Peking University, Beijing, 100871, P.R. China
{ctf,zwm,hujf}@pku.edu.cn

³ Beijing Key Laboratory of Internet Culture and Digital Dissemination Research,
Beijing Information Science and Technology University, Beijing, 100101, P.R. China
lxq@bistu.edu.cn

Abstract. Previous research showed that Kalman filter based human-computer interaction Chinese word segmentation algorithm achieves an encouraging effect in reducing user interventions. This paper designs an improved statistical model for ancient Chinese texts, and integrates it with the Kalman filter based framework. An online interactive system is presented to segment ancient Chinese corpora. Experiments showed that this approach has advantage in processing domain-specific text without the support of dictionaries or annotated corpora. Our improved statistical model outperformed the baseline model by 30% in segmentation precision.

Keywords: Word Segmentation, Human-Computer Interactive System, Kalman Filter, Ancient Chinese Corpus Processing.

1 Introduction

Since Chinese text is written without natural delimiters such as whitespaces, word segmentation is the essential first step in Chinese language processing [1]. Over the past two decades, various methods have been developed to address this issue [2–7]. Generally, supervised statistical learning methods are more robust in processing unrestricted texts than the traditional dictionary-based methods.

However, in some domain-specific applications, for example ancient Chinese text processing, there is neither enough annotated homogeneous corpora for training a reliable statistical model, nor a sufficient lexicon. Under these circumstances, unsupervised methods are preferred to utilize the linguistic knowledge derived from the raw corpus itself. Many researches also explores human-computer interactive segmentation process, enabling users to add expert knowledge to the system

* To whom all correspondence should be addressed.

[8, 9]. Since the criteria of word segmentation is sometimes dependent on users, interactive segmentation is reasonable [10].

Human-computer interactive approaches enables users to review and proof-read the raw segmentation result produced by the statistical model. Zhu et al. proposed a Kalman filter based human-computer interactive learning model for segmenting Chinese texts depending upon neither lexicon nor any annotated corpus [11]. This approach enables users to observe and intervene the segmentation results, while the segmenter learns and adapts to these knowledge iteratively. At the end of this procedure a segmentation result that fully matches the demand of the user is returned.

This paper devises an improved model for ancient Chinese word segmentation, and uses the Kalman filter model by Zhu et al. to implement a practical system for human-computer ancient Chinese text processing.

The rest of this paper is organized as follows. The next section reviews related work. Our statistical model is introduced in Section 3. Section 4 briefly reviews the Kalman filter based approach. In Section 5, we presents the design of our practical segmentation system for ancient Chinese texts. In Section 6, the evaluation is presented, and the final section concludes this paper and discusses possible future work.

2 Related Work

Unsupervised word segmentation is generally based on some predefined criteria, such as *mutual information* (mi), to recognize a substring as a word. Sproat and Shih studied comprehensively in this direction using mutual information [12]. Many successive research applied mutual information with different ensemble methods [13, 14]. Sun et al. designed an algorithm based on the linear combination of mi and *difference of t-score* (dts)[15]. Other criteria like *description length gain* [16], *assessor variety* [17] and *branch entropy* [18] are also explored. Shi et al. adopted conditional random fields to generate a unified process for word segmentation and POS-tagging on pre-Qin ancient Chinese texts [19].

Any automatic segmentation has limitations and is far from fully matching the particular need of users. Thus human-computer interactive strategies are explored to allow users to bring their linguistic knowledge into the segmenter by intervening the segmentation process. Wang et al. developed a sentence-based human-computer interaction inductive learning method [8]. Feng et al. proposed a certainty-based active learning segmentation algorithm, which uses an EM (Expectation Maximization) algorithm to train an n -gram language model in an unsupervised learning framework [20]. Li and Chen further explored a candidate words based human-computer interactive segmentation strategy [21].

The Kalman filter [22] is an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements. Recent researches have introduced Kalman filter model to promote user experience of Internet applications, by estimating click-through rate (CTR) of available articles (or other objects on web pages) in near real-time for news display systems

[23]. Zhu et al. applied Kalman filter model to learn and estimate user intentions in their human-computer interactive word segmentation framework [11].

3 Statistical Model

3.1 Baseline Model

Sun et al. proposed *difference of t-score (dts)* [3] as a useful complement to *mutual information (mi)* and designed a compound statistical measure based on the linear combination of *mi* and *dts*, named *md* [15].

$$mi^*(x, y) = \frac{mi(x, y) - \mu_{mi}}{\sigma_{mi}}, \quad (1)$$

$$dts^*(x, y) = \frac{dts(x, y) - \mu_{dts}}{\sigma_{dts}}, \quad (2)$$

$$md(x, y) = mi^*(x, y) + \lambda \cdot dts^*(x, y), \quad (3)$$

λ is set as an empirical value 0.6 in Sun's paper; $mi(x, y)$ is the normalized mutual information of any given bigram xy , and $dts(x, y)$ is the normalized difference of t-score of bigram xy . Given any bigram xy , in terms of $md(x, y)$ and a threshold Θ , whether this bigram be combined or separated can be determined — when $md(x, y)$ is greater than Θ , the bigram xy is marked as *combined*; otherwise, it is marked as *separated*.

There exists a possible optimization scheme when a local minimum or maximum of md appears [3]. Consider a Chinese character string $wxyz$. If $md(x, y) > md(w, x)$ and $md(x, y) > md(y, z)$, $md(x, y)$ is called a *local maximum*. *Local minima* follow a similar definition. It can be seen that even a $md(x, y)$ of a local maximum does not reach the threshold Θ , xy may still be combined, while if $md(x, y)$ of a local minimum is greater than Θ , xy is more likely to be separated despite its md value. To reflect this kind of tendency, we increase the md values at local maxima by a constant s , and decrease the md values at local minima by s .

This statistical model will be used as a baseline model in further discussions.

3.2 Improved Statistical Model

For bigrams with a smaller number of occurrences, the statistical measure of mutual information (mi) is not reliable. We define a weight for mutual information, i.e.

$$w(x, y) = \log_2(f(x, y) + 1), \quad (4)$$

where $f(x, y)$ is the frequency of bigram xy in the corpus.

Additionally, some proper nouns (e.g. names of people or places) tends to occur only in several adjacent paragraphs or chapters. This rendered the mutual

information of these words low, resulting in these words to be judged as *separate*. If a bigram recurs frequently, i.e. clumps in context, it is more likely to be a content-bearing word [24, 25]. We define a *bigram recurrence* to measure this tendency. Define *bigram recurrence* as

$$br(x, y) = \log_l f_l(x, y), \quad (5)$$

where l is length of context chosen; and f_l is the frequency of bigram xy in context window of length l .

Combine the baseline model and the measures we define above, we define

$$A(x, y) = \lambda_i w(x, y) mi^*(x, y) + \lambda_t dts^*(x, y) + \lambda_r br^*(x, y), \quad (6)$$

where br^* denotes the normalized version of br , and λ_i, λ_t and λ_r are coefficients. If $A(x, y)$ is greater than a threshold Θ , xy is judged as *combined*; otherwise, it is judged as *separated*. We call this function as the A feature. A feature combined global measurements such as mi , as well as local measurement br which is dependent on contexts.

These parameters are trained using an annotated version of *Annals of the Five Emperors, Records of the Great Historian* (《史记·五帝本纪》). These values are chosen as

$$\lambda_i = 0.43, \lambda_t = 1.0, \lambda_r = 0.37, \Theta = 1.0. \quad (7)$$

The local minima and maxima optimization described in Section 3.1 is also exploited in this improved model.

3.3 Structural Words Optimization

In Classical Chinese, some structural words seldom form words with other characters. These characters are judged as single-character words directly in our model. Structural words chosen in this paper includes the following characters:

而, 何, 乎, 乃, 其, 且, 若, 所, 为, 焉, 也, 以, 因, 于, 与, 则, 者, 之, 弗, 莫, 不, 哉, 矣, 又, 已

For example, in character sequence xyz , if y belongs to the structural word set above, the values $A(x, y)$ and $A(y, z)$ are all set to be below the threshold value Θ so that bigram xy and yz are both judged as *separated*.

4 Kalman Filter Model

Zhu et al. developed a human-computer interactive learning word segmentation algorithm using Kalman filters [11]. This model is equipped with a Kalman filter to make it learn and estimate user intentions from the interventions (which may contain noise) for each bigram. The linguistic knowledge is gradually accumulated from the process of user interactions, and eventually, a segmentation result that fully matches the need of the user (or with an accurate rate of 100% by

manual judgement) is returned within limited times of interventions. A basic assumption is that each bigram (of different characters) is independent, i.e., if the state of one bigram is modified, states of other bigrams are not affected.

In this section, we adapt the Kalman filter model by Zhu et al. to the improved statistical model described in Section 3. For simplicity, we focus on a specific bigram xy .

4.1 Process State

A time step is defined as a manual judgement to the segmentation result of a bigram. Since that the system is viewed as a human interaction process, it can also be mapped to a time series process. Given a bigram xy , we assume the statistical measure A in the corpus follows a stable Gaussian distribution $N(\mu, \sigma^2)$, where μ and σ^2 are the expectation and variance of A respectively. We define x_t is the *system state* of the the A feature of a specific bigram at time t . the state of time $t + 1$ is estimated upon time t :

$$\hat{x}_{t+1|t} = \hat{x}_t + w_t , \quad (8)$$

where w_t represents the uncertainty (i.e. noise) of this prediction at time t which follows a normal distribution. This distribution is formulated as

$$w_t \sim N(0, Q_t) , \quad (9)$$

where Q_t is the autocovariance of the bigram at time t . It can be calculated as

$$Q_t = E[(\hat{x}_{t-2} - \mu_{t-2})(\hat{x}_{t-1} - \mu_{t-1})] , \quad (10)$$

where

$$\mu_t = E[\hat{x}_t] \text{ for each } t . \quad (11)$$

Kalman filter also predicts the variance of the state change, which is

$$P_{t+1|t} = P_t + Q_t , \quad (12)$$

where P_t represents the estimation of the state variance at time t .

4.2 Measurements and States Update

Since the system is human-computer interactive, a measurement system that maps manual judgements to a continuous space of A feature is introduced. Apparently, some uncertainty (i.e. observation noise) is inevitable, and we assume that it follows a normal distribution. To guarantee that the mapped values corresponds to the manual judgments, we take a high confidence interval (for example 99%). The system measurements z_t of the true state x_t is assumed to be generated according to

$$z_t = x_t + v_t , \quad (13)$$

where v_t is the uncertainty of observations which is assumed to be a Gaussian white noise R_t . After the observation, The Kalman filter will update the prediction of next state using a Kalman gain [22]:

$$K_t = \frac{P_{t+1|t}}{P_{t+1|t} + R_{t+1}}. \quad (14)$$

The state prediction of time $t + 1$ is updated as

$$\hat{x}_{t+1} = \hat{x}_{t+1|t} + K_t(z_{t+1} - \hat{x}_{t+1|t}), \quad (15)$$

and the updated state variance is estimated as

$$P_{t+1} = (1 - K_t)P_{t+1|t}. \quad (16)$$

Then, this updated state can be used to segment next time this bigram appears in the corpus.

5 Human-Computer Interactive System

In our practical system¹, the user first load the raw corpus into the system. The system will first segment the whole text using the improved statistical model elaborated in Section 3. The interface of the system after loading the raw corpus is shown in Figure 1.

卷一 五帝本纪第一 黄帝 帝者，少典之子，姓公孙， 名曰轩辕。生而神灵，弱而 能言，幼而徇齐，长而敦敏 ，成而聪明。轩辕之时，神 农氏世衰。诸侯相侵伐，暴 虐百姓，而神农氏弗能征。 於是轩辕乃习用干戈，以征 不享，诸侯咸来宾从。而蚩 尤最为暴，莫能伐。炎帝欲 侵陵诸侯，诸侯咸归轩辕。	Pending:	Contexts:
<div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> Flip to next page Save segmented text Save acquired knowledge </div> <p>Current page: 0 ; Characters processed: 0 Interfered bigrams: 0 ; Current page correct % ; Overall page correct %:</p>		

Fig. 1. Snapshot of our system just after loading *Annals of the Five Emperors, Records of the Grand Historian* (《史记·五帝本纪》)

Every bigram's status — whether *combined* or *separated* — can be modified by a click on the symbol between the two characters of a bigram. Modified bigrams

¹ The system can be found at

http://klcl.pku.edu.cn/clr/ccsegweb/kalman_segmenter.aspx.

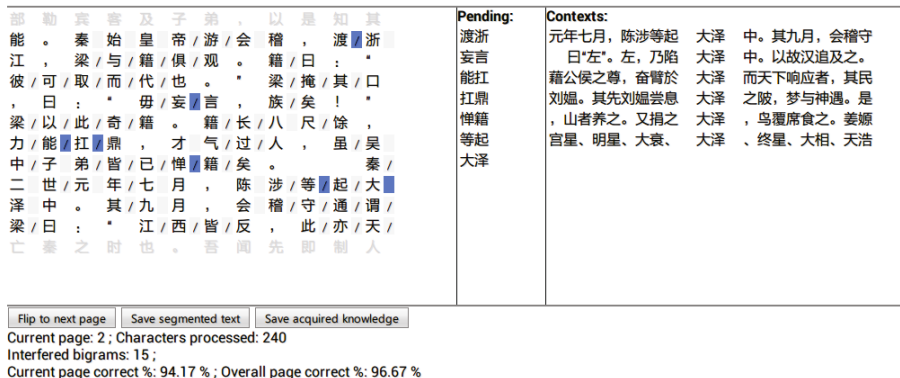


Fig. 2. Snapshot of our system while segmenting *Annals of Xiang Yu, Records of the Grand Historian* (《史记·项羽本纪》)

will be marked with a different color. These clicks act as user input to the system. When the user flips to the next page, pending user interventions will be applied to Kalman filters of these intervened bigrams. Different contexts of the current modified bigram is shown on the right panel of the interface, from which users can check the meaning of this bigram under different contexts.

These features of the system is shown in Figure 2.

During the segmentation process, the system keeps track of the changes the user made, hence it is able to produce better segmentation results as the human-computer interaction progresses. Users can save the current segmentation result and the states of the Kalman filters at any time.

6 Experiments

In this section, we conducted several experiments to evaluate our model. Firstly, we verified the improvement after introducing our new statistical model. Secondly, we verified the effectiveness of Kalman filter model in reducing human effort. The ancient Chinese corpus used for experiments are chapters from *Records of the Grand Historian* (《史记》) and *History of Song* (《宋史》).

As there is no standard specification for ancient Chinese segmentation, we used experts to segment *Annals of Xiang Yu, Records of the Grand Historian* (《史记·项羽本纪》, abbreviated as *Xiang Yu*, approximately 11000 characters) and a part of *Annals of Taizu I, History of Song* (《宋史·本纪第一·太祖一》, abbreviated as *Taizu I*, approximately 2000 characters) as test corpora.

6.1 Improved Statistical Model

In this part, we verified the effectiveness of our improved statistical model without the Kalman filter based human-computer interaction process. Thus, our

improved statistical model acts as an automatic segmenter without the human-computer interaction process. The baseline model used for comparison is the model by Sun et al. [15].

To evaluate the performance of these models, we use the precision and recall rate, as well as the *accuracy of segmentation* (abbreviated as ‘Accuracy’ in this paper) described by Sun et al. in [3]. It is defined as

$$\text{Accuracy}[\%] = \frac{\# \text{ of locations being correctly marked}}{\# \text{ of locations in corpus}} \times 100\% . \quad (17)$$

Corpus for tests are the aforementioned *Xiang Yu* and *Taizu I*. The results are shown in the following two tables.

Table 1. Different measures for *Xiang Yu*

	Accuracy	Precision	Recall
Sun’s Approach	78.18%	54.71%	59.21%
Our model	90.79%	86.94%	80.55%

Table 2. Different measures for *Taizu I*

	Accuracy	Precision	Recall
Sun’s Approach	74.57%	45.60%	55.45%
Our model	88.35%	75.71%	66.44%

From these tables above, it can be seen that our model significantly outperformed Sun’s model because Sun’s model is more suitable to handle contemporary Chinese texts, while our model is optimized on ancient Chinese texts. Our model achieved an improvement of more than 30% in segmentation precision; and achieved an improvement of about 13% ~ 14% in terms of accuracy of segmentation.

Since our model is trained using a text excerpt from *Records of the Grand Historian* (《史记》), *Xiang Yu* is a homogeneous corpus, while *Taizu I* is a heterogeneous corpus. On homogeneous text such as *Xiang Yu*, our model yields a satisfactory result, achieving 86.94% in precision and 80.55% in recall. On heterogeneous text *Taizu I*, a text written more than 1000 years after *Records of the Grand Historian* (《史记》) is completed, the result was still acceptable.

6.2 Kalman Filter Model

In this part, we simulated the human-computer interaction by using the correct segmentation text as input to the model so as to evaluate the performance of the Kalman filter model. We adopted the *binary prediction rate* (BPR) described by Zhu et al. [11] to quantify the conformity of the prediction in the model with user intention,

$$\text{BPR}[\%] = \frac{\# \text{ of correct predictions}}{\# \text{ of all predictions}} \times 100\% . \quad (18)$$

Two models were compared in this section. One is the approach discussed in Section 6.1, i.e. our improved statistical model without the human-computer interaction process is abbreviated as *Without learning*), and the other is the Kalman filter integrated approach discussed in Section 4 (abbreviated as *Kalman approach*).

The result of the experiment is shown in Table 3. Corpora used is the same as the previous section.

Table 3. The BPR[%] of different approaches

Corpus	<i>Xiang Yu</i>	<i>Taizu I</i>
Without learning	90.79%	88.35%
Kalman approach	92.38%	88.86%

From this experiment, it can be seen that on homogeneous text such as *Xiang Yu*, Kalman filter based human-computer interactive model outperformed the baseline statistical value by about 1.6%. On *Taizu I*, the improvement was insignificant because the text is rather short (approximately 2000 characters).

7 Conclusions and Future Work

Previous research showed that Kalman filter based human-computer interaction Chinese word segmentation algorithm achieves an encouraging effect in reducing user interventions. This paper designs an improved statistical model for ancient Chinese texts, and integrates it to the Kalman filter based framework, resulting in a practical system. Experiments revealed that our improved statistical model significantly outperforms the baseline model, and the Kalman filter approach achieves a notable improvement in reducing human efforts.

Our future work will focus on establishing an interactive bootstrapping segmentation system with an accumulating dictionary.

Acknowledgments. This work is partially supported by Open Project Program of the National Laboratory of Pattern Recognition (NLPR) and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201102).

References

1. Liang, N.Y.: CDWS: An Automatic Word Segmentation System for Written Chinese Texts. *Journal of Chinese Information Processing* 2(2), 44–52 (1987) (in Chinese)
2. Nie, J.Y., Jin, W., Hannan, M.L.: A Hybrid Approach to Unknown Word Detection and Segmentation of Chinese. In: *Proceedings of the International Conference on Chinese Computing*, pp. 326–335 (1994)

3. Sun, M., Shen, D., Tsou, B.K.: Chinese Word Segmentation Without Using Lexicon and Hand-Crafted Training Data. In: COLING/ACL 1998, pp. 1265–1271 (1998)
4. Luo, X., Sun, M., Tsou, B.K.: Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information. In: COLING 2002, pp. 1–7 (2002)
5. Zhang, H.P., Liu, Q., Cheng, X.Q., Yu, H.K.: Chinese Lexical Analysis Using Hierarchical Hidden Markov Model. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, pp. 63–70 (2003)
6. Peng, F., Feng, F., McCallum, A.: Chinese Segmentation and New Word Detection Using Conditional Random Fields. In: COLING 2004, pp. 23–27 (2004)
7. Goldwater, S., Griffiths, T.L., Johnson, M.: Contextual Dependencies in Unsupervised Word Segmentation. In: COLING/ACL 2006, pp. 673–680 (2006)
8. Wang, Z., Araki, K., Tochinai, K.: A Word Segmentation Method with Dynamic Adapting to Text Using Inductive Learning. In: Proceedings of the First SIGHAN Workshop on Chinese Language Processing, pp. 1–5 (2002)
9. Li, M., Gao, J., Huang, C., Li, J.: Unsupervised Training for Overlapping Ambiguity Resolution in Chinese Word Segmentation. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, pp. 1–7 (2003)
10. Sproat, R., Gale, W., Shih, C., Chang, N.: A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. *Computation Linguistics* 22(3), 377–404 (1996)
11. Zhu, W., Sun, N., Zou, X., Hu, J.: The Application of Kalman Filter Based Human-Computer Learning Model to Chinese Word Segmentation. In: Gelbukh, A. (ed.) *CICLing 2013, Part I. LNCS*, vol. 7816, pp. 218–230. Springer, Heidelberg (2013)
12. Sproat, R., Shih, C.: A Statistical Method for Finding Word Boundaries in Chinese Text. In: *Computer Processing of Chinese and Oriental Languages*, pp. 336–351 (1990)
13. Chien, L.F.: Pat-Tree-Based Keyword Extraction for Chinese Information Retrieval. *ACM SIGIR Forum*, 50–58 (1997)
14. Yamamoto, M., Kenneth, C.W.: Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. *Computer Linguistics* 27(1), 1–30 (2001)
15. Sun, M., Xiao, M., Tsou, B.K.: Chinese Word Segmentation without Using Dictionary Based on Unsupervised Learning Strategy. *Chinese Journal of Computers* 27(6), 736–742 (2004) (in Chinese)
16. Kit, C., Wilks, Y.: Unsupervised Learning of Word Boundary with Description Length Gain. In: Proceedings of the CoNLL 1999 ACL Workshop, pp. 1–6 (1999)
17. Feng, H., Chen, K., Deng, X., Zheng, W.: Accessor Variety Criteria for Chinese Word Extraction. *Computation Linguistics* 30(1), 75–93 (2004)
18. Jin, Z., Tanaka-Ishii, K.: Unsupervised Segmentation of Chinese Text by Use of Branching Entropy. In: COLING/ACL 2006, pp. 428–435 (2006)
19. Shi, M., Li, B., Chen, X.: CRF Based Research on a Unified Approach to Word Segmentation and POS Tagging for Pre-Qin Chinese. *Journal of Chinese Information Processing* 24(2), 39–45 (2010) (in Chinese)
20. Feng, C., Chen, Z., Huang, H., Guan, Z.: Active Learning in Chinese Word Segmentation Based on Multigram Language Model. *Journal of Chinese Information Processing* 20(1), 50–58 (2006) (in Chinese)
21. Li, B., Chen, X.: A Human-Computer Interaction Word Segmentation Method Adapting to Chinese Unknown Texts. *Journal of Chinese Information Processing* 21(3), 92–98 (2007) (in Chinese)
22. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* 82(1), 35–45 (1960)

23. Agarwal, D., Chen, B.C., Elango, P., Motgi, N., Park, S.T., Ramakrishnan, R., Roy, S., Zachariah, J.: Online Models for Content Optimization. In: Proceedings of NIPS 2008, pp. 17–24 (2008)
24. Liu, Z., Sun, M.: Web-Based Automatic Detection for IT New Terms. In: Proceedings of the 9th China National Conference on Computational Linguistics, pp. 515–521 (2007)
25. Bookstein, A., Klein, S.T., Raita, T.: Clumping Properties of Content-bearing Words. *Journal of the American Society for Information Science* 49(2), 102–114 (1998)

Chinese Word Segmentation with Character Abstraction

Le Tian, Xipeng Qiu*, and Xuanjing Huang

School of Computer Science, Fudan University, China
xpqiu@fudan.edu.cn

Abstract. Chinese word segmentation is an important and necessary problem to analyze Chinese texts. In this paper, we focus on the primary challenges in Chinese word segmentation: low accuracy of out-of-vocabulary word. To resolve this difficult problems, we group the “similar” characters to generate more abstract representation. Experimental results show that character abstraction yields a significant relative error reduction of 24.83% in average over the state-of-the-art baseline.

1 Introduction

Although words are the basic language units in Chinese, Chinese sentences consist of the continuous sequence of characters (called Hanzi) without no space between words. Therefore, word segmentation is a necessary initial step to process the Chinese language. Previous research shows that word segmentation models trained on labeled data are reasonably accurate.

Currently, the state-of-art Chinese word segmentation (CWS) methods are mostly based on sequence labeling algorithm with word-based or character-based features[16,12,14,1]. These methods use the discriminative model with millions of overlapping binary features.

Recent works have tended to be “feature engineering” by trying various well-designed features to obtain the best performance. Intuitively, the complex features can give more accurate prediction than simple features, and these methods often perform remarkably well. However, they still suffer from the out-of-vocabulary (OOV) words (namely, unknown words) problem. Although the accuracy of OOV can be improved greatly by the character-based methods [16], it is still significantly lower than the accuracy of in-vocabulary (IV) words.

To deal with this problem, we wish to merge the characters, which are used in paradigmatical similar way, into abstract representation.

According to our statistics, the distribution of the characters is very skew and is subject to Zipf’s law. As reported in [18], though modern Chinese character sets normally include about 10,000-20,000 characters, most of them are rarely used in everyday life. Typically, 2,500 most used Chinese characters can cover 97.97% text, while 3,500 characters can cover 99.48% text.

* Corresponding author.

The sparsity has a large potential to compress the space of the characters in an abstraction way, which can also bridge the gap between high and low frequency characters.

Although, there are some works [10] to use character clustering, such as Brown algorithm [2]. However, Brown algorithm classifies all the same characters into a single cluster, but Chinese characters may have many senses, hard clustering algorithm such as the Brown algorithm may not be able to deal with multiple senses gracefully. Moreover, Brown algorithm generally tends to cluster the characters which significantly occur together in text. The characters in same cluster have syntagmatical similarity, not paradigmatical similarity. [10] also reports that the features with these clusters do not improve performance on CWS.

In this paper, we propose a Chinese word segmentation method with abstraction on character levels. In Chinese, some characters are semantically or paradigmatically similar. We abstract characters into their semantic concept space. We propose a semi-supervised k-means clustering method to cluster the similar characters to the same class according their context. The map function is learned from large-scale raw texts, which can capture paradigmatic similarity among characters. Our approach yields a relative error reduction of 24.83% and an improvement of OOV recall of 34.92% in average with character abstraction over the baseline.

The rest of the paper is organized as follows: We first introduce the related works in section 2, then we describe the background of character-based word segmentation in section 3. Section 4 presents our character abstraction method. The experimental results are manifested in section 5. Finally, we conclude our work in section 6.

2 Related Works

Character abstraction is very similar to word cluster in English.

[10] uses character clustering features derived using the Brown algorithm [2] and finds that they do not improve performance on CWS. One problem might be that Chinese characters have many more senses than English words, so a hard clustering algorithm such as the Brown algorithm may not be able to deal with multiple senses gracefully.

[9] use word clustering features from a soft word clustering algorithm which can improve performance of CWS.

More broadly, [15] evaluate Brown clusters, word embeddings [4], and HLBL [11] embeddings of words on other sequence labeling tasks (Named Entity Recognition and chunking) and find that each of the three word representations improves the accuracy of these baselines.

3 Discriminative Character-Based Word Segmentation

We use discriminative character-based sequence labeling for word segmentation. Each character is labeled as one of {B, I, E, S} to indicate the segmentation. {B, I, E} represent *Begin*, *Inside*, *End* of a multi-character segmentation respectively, and S represents a *Single* character segmentation.

Sequence labeling is the task of assigning labels $\mathbf{y} = y_1, \dots, y_n$ to an input sequence $\mathbf{x} = x_1, \dots, x_n$. Given a sample \mathbf{x} , we define the feature $\Phi(\mathbf{x}, \mathbf{y})$. Thus, we can label \mathbf{x} with a score function,

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} S(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})), \quad (1)$$

where \mathbf{w} is the parameter of score function $S(\cdot)$. The feature vector $\Phi(\mathbf{x}, \mathbf{y})$ consists of lots of overlapping features, which is the chief benefit of discriminative model. We use online Passive-Aggressive (PA) algorithm [5,6] to train the model parameters. Following [3], the average strategy is used to avoid the overfitting problem.

4 Character Abstraction: Learning Character Semantic Concepts

To bridge the gap between high and low frequency character, we first map the characters with same semantic concepts into a single cluster. Different with English letters, Chinese characters are associated with full or partial semantic concepts. For example, the characters “鸡(chickens)”, “鸭(ducks)” and “鹅(geese)” have the same concept “fowl”. They are used in the same way to compose words with other characters, such as “头(head)”, “爪(feet)” and “肉(meat)”. Since characters that appear in similar contexts (especially surrounding words) tend to have similar meanings, we can use clustering technologies to find the semantic concepts from large-scale texts.

The Chinese character clustering is similar to English word clustering. Both of them partitions sets of words/characters into subsets of semantically similar words/characters.

Brown cluster algorithm [2] is a popular algorithm of word cluster which derives a hierarchical clustering of words from unlabeled data.

Table 1 shows the top clusters derived with Brown algorithm. Besides “又/却”, “刘/杨” and “八/七”, the other clusters are not what we expected. The characters in each cluster are often collocation relations. These clusters are helpful for other NLP tasks, such as text classification, but may be misleading in CWS.

4.1 Semi-supervised K-means Cluster

To avoid the shortcomings of Brown clustering algorithm, we propose a semi-supervised K-means clustering method to map each character to its corresponding concepts based on its context.

Table 1. Top Clusters derived with Brown algorithm. (Each column represents a cluster.)

编	康	引	矿	阿	截	汉	又	喜	微	刘	八	圣	必	经	国	北	们	2	资	公
辑	健	吸	煤	伊	甚	武	却	欢	博	杨	七	诞	须	济	中	京	我	月	投	司

Different with unsupervised clustering methods, we first use HowNet Knowledge Database[7] as a initial guide for semantic concepts, then we use k-means algorithm to cluster on large-scale unlabeled texts.

HowNet gives the means not only for each word but also for each Chinese character. We extract all single characters and the corresponding semantic concepts from HowNet and categorize them by their semantic concepts. Each category represents a different semantic concept or meaning. There are 7,117 characters and 3,666 categories in total. 2,979 characters belong to more than one category. Among them, “打” has the most meanings and belongs to 32 different categories, such as “dozen”, “draw”, “beat”, “build”, “call” and so on.

The detailed statistics are shown in Table 2.

Table 2. Categories of Characters from HowNet

Number of Characters	7,117
Number of Categories	3,666
Average Number of Characters per Category	3.99
Average Number Categories of per Character	2.06

Although each character can belong to more than one category, it can has one meaning in certain context. So we need map different occurrences of a character to different categories based on their different contexts. We use k-means algorithm [8] to automatically learn the map function from large scale texts.

We set the number of clusters to 3,666, which is same to the number of semantic categories defined in HowNet.

The initial center for each cluster m_i is calculated by

$$m_i = \frac{\sum_x \sum_{x \in CH_i} \mathbf{f}(x)}{\sum_x \sum_{x \in CH_i} \mathbf{1}} \quad (2)$$

where x is every occurrence of character and $\mathbf{f}(x)$ means the context feature of x ; CH_i represents the i^{th} category defined in HowNet. All the feature vectors are extracted from unlabeled data.

In order to better consider the context information, we use the previous, succeeding and the union of them as features. For example: the features of the character “鸡” in sequence “吃鸡肉” will be $\{-1: \text{吃}, 1: \text{肉}, \text{吃肉}\}$.

Then we re-assign each occurrence of character to the nearest cluster. The distance we used here is Euclidean distance. The cluster center will be updated when it changes. We make a restriction that the assigned cluster for each character must be one of its categories defined in HowNet.

We use large scale unlabeled corpus from web pages collected by Sogou¹ to learn the cluster centers. The corpus contains 1,060,471,497 characters and has 9,851 unique characters. For the characters which are not included in HowNet, we classify them into “unknown” category.

When using in CWS, we find the category for each character x by

$$c = \arg \max_i \|f(x) - m_i\|^2 \quad (3)$$

Table 3 shows some examples for character abstraction. When we represent the character with its category, we can avoid the problem of data sparsity on some level.

Table 3. Examples for Character Abstraction

Abstraction Representation	Sequences of Characters
$C_{Animal}-C_{BodyPart}$	猪头, 狗头, 鸡脚
$C_{Number}-C_{Unit}$	5 毫, 五尺, 3 寸, 9 码, 壹分, 7 里, 八米
$C_{Surname}$	覃 (经理), 温 (总理), 冯 (先生), ...

5 Experiments

5.1 Dataset

All our experiments are conducted on the corpora provided by CIPS-SIGHAN-2010[17]. This dataset is well known and widely adopted. The training corpus which contains one month data of the People’s Daily in 1998 was provided by Peking University. There are four domains in the testing data: Literature (L), Computer (C), Medicine (M) and Finance (F). One main reason we use these corpora is that it addresses the ability of word segmentation for out-of-domain text.

We implement our system based on FudanNLP [13], a toolkit for Chinese natural language processing.

We first evaluate the performance with our character abstraction method.

1. **B:** The baseline method. We use usually the commonly used features in CWS. The form of features is shown in Table 4, where C represents a Chinese character, and T represents the character-based tag. The subscript i indicates its position related to the current character.

¹ <http://www.sogou.com/labs/dl/ca.html>

Table 4. Traditional Feature Templates

$C_i, T_0 (i = -2, -1, 0, 1, 2)$
$C_i, C_j, T_0 (i, j = -2, -1, 0, 1, 2 \text{ and } i \neq j)$
C_{-1}, C_0, C_1, T_0
T_{-1}, T_0

Table 5. Performances of Different Methods

	Methods	R	P	F1	R_{OOV}	R_{IV}
L	B	0.905	0.911	0.908	0.540	0.932
	B+C	0.905	0.915	0.910	0.546	0.932
	B+CA	0.913	0.921	0.917	0.618	0.935
C	B	0.864	0.784	0.822	0.388	0.950
	B+C	0.881	0.907	0.894	0.520	0.946
	B+CA	0.914	0.921	0.918	0.704	0.952
M	B	0.899	0.894	0.897	0.594	0.937
	B+C	0.897	0.899	0.898	0.602	0.934
	B+CA	0.905	0.907	0.906	0.657	0.936
F	B	0.909	0.905	0.907	0.506	0.948
	B+C	0.911	0.921	0.916	0.528	0.948
	B+CA	0.930	0.934	0.932	0.674	0.955

- B+C**: Besides the features used in baseline method, we use the character type features directly extracted from HowNet. For the character belonging to multiple types, we use all its types directly. This method can be regarded as the manually character clustering and is more accurate than Brown algorithm.
- B+CA**: Besides the features used in baseline method, we use our proposed character abstract features. The reason we use the character features is that there are some culture-specific words, common saying or idioms. These words is used regardless of character type, such as “不管三七二十一 (regardless of the consequence)” , “由此可见 (thus it can be seen)” , etc.

The results are shown in 5, which shows the information of character type can improve the performances. While the improvement is very limited to simply use these information **B+C**, our method (**B+CA**) achieves large improvements on the baseline (**B**) and yields the relative error reductions of 9.78%, 53.93%, 8.74% and 26.88% respectively on four datasets. The average relative error reduction is 24.83%. Meanwhile, the recalls OOV words are also improved by 14.44%, 81.44%, 10.61% and 33.20% respectively. The average improvement of OOV recalls is 34.92%.

Table 6 gives the numbers of nonzero parameters of models trained by different methods. We can see that our method (**B+CA**) uses fewer parameters than the baseline, which indicates the character abstraction can merge the characters used similarly and results to reduction of actually active features.

Table 6. Number of Nonzero Parameters

Methods	Number
B	2,022,396
B+C	2,683,524
B+CA	1,501,705

5.2 Analysis

We can see from the above experiments that The character abstraction can still boost the performance with less actually active features. However, we also found a number of inconsistent or irrational annotations in segmentation both in the training and the test data. For example, “建设银行(Construction Bank)” is segmented while “中国银行(Bank of China)” is used a word. These inconsistent or irrational annotations may have more impact for abstraction based method than character-based method because they can influence the process of feature abstraction.

6 Conclusion

In this paper, we focus on the challenges in Chinese word segmentation: low accuracy of out-of-vocabulary word. The experiments have shown that: abstract representation can improve the performance over the baseline. In future work, we would also like to investigate the other methods for character abstraction and we believed that good abstract features can boost the performance of CWS.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments. This work was funded by NSFC (No.61003091 and No.61073069).

References

1. Andrew, G.: A hybrid markov/semi-markov conditional random field for sequence segmentation. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 465–472. Association for Computational Linguistics (2006)
2. Brown, P., Desouza, P., Mercer, R., Pietra, V., Lai, J.: Class-based n-gram models of natural language. *Computational Linguistics* 18(4), 467–479 (1992)
3. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (2002)
4. 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, pp. 160–167. ACM (2008)
5. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research* 3, 951–991 (2003)

6. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7, 551–585 (2006)
7. Dong, Z., Dong, Q.: *HowNet and the Computation of Meaning*. World Scientific Publishing Co., Inc., River Edge (2006)
8. Duda, R., Hart, P., Stork, D.: *Pattern Classification*, 2nd edn. Wiley, New York (2001)
9. Li, W., McCallum, A.: Semi-supervised sequence modeling with syntactic topic models. In: *Proceedings of the National Conference on Artificial Intelligence*, p. 813 (2005)
10. Liang, P.: *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology (2005)
11. Mnih, A., Hinton, G.: A scalable hierarchical distributed language model. In: *Advances in Neural Information Processing Systems 21*, pp. 1081–1088 (2009)
12. Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: *Proceedings of the 20th International Conference on Computational Linguistics* (2004)
13. Qiu, X., Zhang, Q., Huang, X.: FudanNLP: A toolkit for Chinese natural language processing. In: *Proceedings of ACL* (2013)
14. Sarawagi, S., Cohen, W.: Semi-markov conditional random fields for information extraction. In: *Advances in Neural Information Processing Systems 17*, pp. 1185–1192 (2005)
15. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. *Urbana* 51, 61801 (2010)
16. Xue, N.: Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing* 8(1), 29–48 (2003)
17. Zhao, H., Huang, C., Li, M., Lu, B.: A unified character-based tagging framework for Chinese word segmentation. *ACM Transactions on Asian Language Information Processing (TALIP)* 9(2), 5 (2010)
18. Zhao, H., Liu, Q.: The cips-sighan clp 2010 Chinese word segmentation bakeoff. In: *Proceedings of the First CPS-SIGHAN Joint Conference on Chinese Language Processing* (2010)

A Refined HDP-Based Model for Unsupervised Chinese Word Segmentation

Wenzhe Pei, Dongxu Han, and Baobao Chang

Key Laboratory of Computational Linguistics, Ministry of Education,
Institute of Computational Linguistics, School of Electronics Engineering
and Computer Science, Peking University
{peiwenzhe, handx, chbb}@pku.edu.cn

Abstract. This paper proposes a refined Hierarchical Dirichlet Process (HDP) model for unsupervised Chinese word segmentation. This model gives a better estimation of the base measure in HDP by using a dictionary-based model. We also show that the initial segmentation state for HDP model plays a very important role in model performance. A better initial segmentation can lead to a better performance. We test our model on PKU and MSRA datasets provided by Second Segmentation Bake-off (SIGHAN 2005) [1] and our model outperforms the state-of-the-art systems.

1 Introduction

Chinese word segmentation is a very important component for almost all natural language processing tasks. Although supervised segmentation systems have been widely used, they rely on manually segmented corpora, which are often specific to domain and various kinds of segmentation guidelines. As a result, supervised segmentation systems perform poorly on out-of-domain corpus such as Microblog corpus [2] which contains lots of new words and domain specific words. In order to tackle this problem, unsupervised word segmentation becomes a very important issue. Various kinds of models have been proposed for unsupervised word segmentation task. [3] compared several popular models for unsupervised word segmentation with a unified framework. [4] presented a model based on the Variation of Branching Entropy. [5] proposed an iterative model based on a new goodness algorithm that adopts a local maximum strategy and avoids thresholds.

In this paper, we present an unsupervised word segmentation method which refines the HDP-Based model [6]. This model gives a better estimation of the base measure in HDP by using a dictionary-based model. We also show that the initial segmentation state for HDP model plays a very important role in model performance. A better initial segmentation can lead to a better performance. We test our system on the PKU and MSRA benchmark datasets provided by Second Segmentation Bake-off (SIGHAN 2005) [1] and our method performed better than the state-of-the-art systems.

The remainder of this paper is structured as follows. In section 2, we give an overview of the HDP-based unsupervised word segmentation model. In section

3, we describe our models in detail. Section 4 shows our experiment results on the benchmark dataset. We then conclude the paper with section 5.

2 HDP-Based Unsupervised Word Segmentation

The Dirichlet Process (DP) is a stochastic process used in Bayesian non-parametric models of data. Let H be a distribution called base measure. The DP is a probability distribution, i.e. each draw from a DP is itself a distribution over distributions

$$G \sim DP(\alpha, H)$$

where H is basically the mean of the DP and α can be understood as an inverse variance. We can see that Dirichlet Process can be viewed as an infinite dimensional generalization of Dirichlet distributions.

The Hierarchical Dirichlet Process (HDP) is an extension to DP. It is a non-parametric Bayesian approach to clustering grouped data. It uses a Dirichlet process for each group of data, with the Dirichlet processes for all groups sharing a base distribution which is itself drawn from a Dirichlet process. The process defines a set of random probability measure G_j , one for each group, and a global random probability measure G_0 . The global measure G_0 is distributed as a DP with concentration parameter α and base measure H and the random measure G_j are given by a DP with concentration parameter α_1 and base measure G_0

$$\begin{aligned} G_j &\sim DP(\alpha_1, G_0) \\ G_0 &\sim DP(\alpha, H) \end{aligned}$$

[6] proposed a Bayesian framework for unsupervised word segmentation with HDP. They define a bigram model by assuming each word has a different distribution over the words that follow it, but all these distributions are linked:

$$\begin{aligned} w_i | w_{i-1} = l &\sim G_l \\ G_l &\sim DP(\alpha_1, G_0) \\ G_0 &\sim DP(\alpha, H) \end{aligned}$$

That is, $P(w_i | w_{i-1} = l)$ is distributed according to G_l which is a DP specific to word l . G_l is linked to other DPs by sharing a common base distribution G_0 . The generating process can be represented according to the Chinese Restaurant Franchise (CRF) [7] metaphor. The metaphor is as follows. We have a restaurant franchise with a shared menu G_0 and each restaurant has infinitely many tables. When the $n + 1$ th customer enter the restaurant l , the customer either joins an already occupied table k with probability proportional to the number n_k of customers already sitting there and share the dish, or sits at a new table with probability proportional to α_1 and order a dish from menu G_0 . Choosing dish from menu G_0 is a similar process, we can either choose an already ordered dish j with probability proportional to the number n_j of dishes already been ordered

by all restaurants, or choose a new dish from H with probability proportional to α . In this bigram model, each w_{i-1} corresponds to a restaurant and each w_i is a dish. In practice, we can not observe G_l and G_0 directly because it will be infinite dimensional distribution over possible words. However, we can integrate out G_l and G_0 to get the posterior probability $P(w_i|w_{i-1} = l, h)$, where h is the observed segmentation result:

$$\begin{aligned}
 & P(w_i|w_{i-1} = l, h) \\
 &= \int P(w_i|w_{i-1} = l, G_l)P(G_l|h)dG_l \\
 &= \frac{n_{\langle w_{i-1}, w_i \rangle} + \alpha_1 P(w_i|h)}{n_l + \alpha_1}
 \end{aligned} \tag{1}$$

Here $n_{\langle w_{i-1}, w_i \rangle}$ is the number of occurrences of the bigram $\langle w_{i-1}, w_i \rangle$ in the observed segment h and $P(w_i|h)$ is defined as:

$$\begin{aligned}
 & P(w_i|h) \\
 &= \int P(w_i|G_0)P(G_0|h) \\
 &= \frac{t_{w_i} + \alpha H(w_i)}{t + \alpha}
 \end{aligned} \tag{2}$$

Here t_{w_i} is the total number of tables labeled with w_i , t is the total number of tables and $H(w_i)$ is the prior knowledge of the probability of word w_i .

Given the equation of posterior probability, Gibbs sampling [8] is used for word segmentation by repeatedly sampling the value of each possible word boundary location, conditioned on the current values of all other boundary locations. So each sample is from a set of two hypotheses: current location is either a word boundary or not. For example, let current segmentation result be $\beta c_{i-2}c_{i-1}c_i c_{i+1}c_{i+2}\gamma$, where β and γ are the sequence of words to the left and right of the area under consideration and $c_{i-2}c_{i-1}c_i c_{i+1}c_{i+2}$ forms a word w (Each c corresponds to a Chinese character). If the current sampling location i is a word boundary, the segmentation result would become $\beta w_1 w_2 \gamma$ where $w_1 = c_{i-2}c_{i-1}c_i$ and $w_2 = c_{i+1}c_{i+2}$. Otherwise, the segmentation result would remain the same. Let h_1 the first hypotheses and h_2 be the second. The posterior possibility for Gibbs sampling would be:

$$P(h_1|h^-) = P(w_1|w_l, h^-)P(w_2|w_1, h^-)P(w_r|w_2, h^-)$$

$$P(h_2|h^-) = P(w|w_l, h^-)P(w_r|w, h^-)$$

Here h^- is the current values of all other boundary locations without current position i and w_l (w_r) is the first word to the left (right) of the current word w . After the Gibbs sampling converged, the segmentation result can be obtained according to the word boundary results.

3 Refined Model

In this section, we present our model in detail and show how HDP-based model can be refined to improve the segmentation performance.

3.1 Improved Base Measure

As we can see in equation (1) and (2), the effect of posterior possibility of HDP is in fact a kind of smoothing. The bigram probability is smoothed by backing off to the unigram model and the unigram is smoothed by the base measure H , namely the prior probabilities over words. If the lexicon is finite, we can use a uniform prior $H(w) = \frac{1}{|V|}$. However, as every substring could be a word, the lexicon will be countbaly infinite. So building an accurate H is very important for word segmentation. [6] used a unigram character-based language model. [9] used a uniform distribution over characters dependent on word length with a Poission distribution.

In this paper, we use a dictionary-based model for estimating H . The intuition behind of our method is that given a large segmented corpus, a better estimation of the probability of a word can be obtained by using maximum likelihood estimation which is much more accurate than a simple character-based unigram model. However, in an unsupervised word segmentation task, we do not have a segmented corpus for probability estimation. To get the segmented corpus in an unsupervised way, we can use other unsupervised word segmentation system to segment the corpus. Although this could be inaccurate, substrings that are recognized as words would tend to have a high probability in the segmented corpus. To obtain a better estimation, we use different unsupervised word segmentation models to segment the corpus and merge the results together. Because different models give a different view of what a word is. The substring which is a real word tends to be recognized by all the models thus having a high probability. On the other hand, substring that is not a word tends to appear in none of the models.

As can be seen in Fig.1, we first use different unsupervised word segmentation systems to segment the training corpus. Then the words whose frequency is bigger than a threshold are selected from all the segmentation results and we merge the results to form a dictionary, that is, the frequency of the same word from different results are added up. Given the dictionary of words with their frequency, the base measure H is defined as follows:

$$\begin{aligned}
 H(w_i) &= \gamma P_{ml}(w_i) + (1 - \gamma) P_{smooth}(w_i) \\
 P_{ml}(w_i) &= \frac{C_{w_i}}{\sum_i^{|V|} C_{w_i}} \\
 P_{smooth}(w_i) &= (1 - p_s)^{|w_i|-1} p_s \prod_j p(c_{ij})
 \end{aligned}$$

Here, $P_{ml}(w_i)$ is the maximum likelihood estimation of the word probability from the dictionary and $P_{smooth}(w_i)$ is the base measure defined by [6]. As defined in

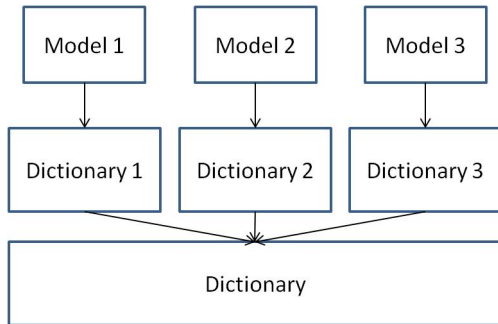


Fig. 1. Dictionary formed by exploiting different unsupervised word segmentation systems

[6], p_s is the probability of generating a word boundary. Thus $(1-p_s)^{|w_i|-1}p_s$ can be seen as the probability of $p(|w_i|)$ where $|w_i|$ is the length of word w_i . $p(c_{ij})$ is the probability of the j th character c_{ij} of word w_i , which can be obtained using maximum likelihood estimation from training data. $P_{ml}(w_i)$ and $P_{smooth}(w_i)$ are interpolated by parameter γ to make a trade-off between the two kinds of probability. As we can see in section 4, this better estimation of base measure H helps improve the model performance.

3.2 Initial State

As we present in section 2, the HDP model iteratively samples the value each possible word boundary location using Gibbs sampling. The procedure of sampling can be viewed as a random search in space of possible segmentation states. [6] use a random segmentation as the initial segmentation state and show that even with random initial state the model can still converge to a good result. However, we believe that a better initial state can lead to a better result and help converge much faster. In our method, we use a state-of-the-art unsupervised word segmentation system to segment the data first and use the segmentation result as a initial segmentation for the HDP model. As we can see in section 4, by using a better initial state, our method obtained a much better result than both the state-of-the-art system and the HDP model with random initial state.

4 Experiment

In this section we test our model on PKU and MSRA datasets released by the Second Segmentation Bake-off (SIGHAN 2005) [1] and make a comparison with previous work.

4.1 Prior Knowledge Used

Concerning unsupervised Chinese segmentation, a problem needs to be clarified is to what extent prior knowledge could be injected into the model. To be an as

strict unsupervised model as possible, no prior knowledge such as word length, punctuation information, encoding scheme could be used. However, information like punctuation can be easily used to improve the performance. The problem is that we could not know what kind of prior knowledge other models used. For example, one might use manually designed regular expression to deal with numbers and dates, but does not list the regular expressions in paper. This makes it difficult to re-implement other models and make a fair comparison. To compare our model with previous models under the same condition, only punctuation information is used in our experiments. Punctuation information can improve the performance, since such information usually unambiguously marks boundary of words. It is very reasonable to use them in unsupervised Chinese segmentation model.

4.2 Model Selection

We randomly selected 2000 sentences from the training data as our development set for parameter tuning. We set $\alpha_1=100$, $\alpha=10$, $\gamma=0.8$, $p_s=0.5$. We used two unsupervised word segmentation model to form the dictionary and give a initial segmentation result as described in section 3. The first model we use is nVBE [4]. It follows Harris’s hypothesis in Kempe [10] and Tanaka-Ishii’s [11] reformulation and base their work on the Variation of Branching Entropy. They improve on [12] by adding normalization and viterbi decoding. This model achieves state-of-the-art results on the Second Segmentation Bake-off (SIGHAN 2005) datasets. The second model we use is based on mutual information. Using mutual information is motivated by the observation of previous work by Hank and Church [13]. If character A and character B have a relatively high MI that is over a certain threshold, we prefer to identify AB as a word over those having lower MI values. We computed the mutual information on the training data. During the segmentation, we separate two adjacent characters to form a word boundary if their MI value is lower than a threshold. The threshold is set to 2.5 in our experiment. Although this model is not the state-of-the-art model, it is easy to implement and do give a different view of what a word is compared with nVBE. We put the training and test data together for segmenting. The word frequency threshold is set to 10 and two segmentations are merged to form the final dictionary.

4.3 Experiment Result

We test our model on the PKU and MSRA datasets released by the Second Segmentation Bake-off (SIGHAN 2005) [1]. We re-implement the nVBE model and the MI model and build our model based on these implementations. All the training data and test data are merged together for segmentation and only the test data are used for evaluation. The overall F-scores of different models are given in Table 1.

We can see that by using a dictionary-based model for estimating the base measure, the HDP model (HDP + dict) achieves a better result although only

Table 1. Comparison of experiment results on PKU and MSRA datasets released by Second Segmentation Bake-off (SIGHAN 2005). ESA corresponds to the model in [5].

Model	PKU	MSRA
HDP	68.7	69.9
nVBE ¹	77.9	78.2
ESA [5]	77.4	78.4
MI	66.1	70.2
HDP + dict	69.2	70.5
HDP + nVBE	79.2	79.4
HDP + MI	72.6	74.4
HDP + nVBE + dict	79.3	79.8

by a small margin. By using the segmentation result of nVBE as the initial segmentation, the HDP model (HDP+nVBE) gets a much better result than both the original HDP model and the nVBE model. Compared with nVBE, the F-score increases by 1.3% on PKU corpora and 1.2% on MSRA corpora. The HDP model with initial segmentation by MI (HDP+MI) also obtained a better result but not as well as HDP+nVBE model. This shows that the initial segmentation do play an important role in the model performance. A better initial segmentation tends to lead to a better performance. What’s more, we find that with a better initial segmentation, the algorithm converges much faster than ordinary HDP. The HDP+nVBE converged after about 50 iterations while ordinary HDP needed 1000 iterations to converge. This saves a lot of time as sampling on a large dataset can be quite slow. The best model (HDP+nVBE+dict) is obtained by using the initial segmentation of nVBE and giving better estimation of base measure with the dictionary-based model. Many errors are related to dates, Chinese numbers and English words. We believe that with a better preprocessing our model can achieve a much better result.

5 Conclusion

In this paper, we proposed a refined HDP model for unsupervised Chinese word segmentation. The refined HDP model uses a better estimation of base measure and replaces the random initial segmentation with a better one by exploiting other state-of-the-art unsupervised word segmentation systems. The refined HDP model achieves much better result than the state-of-the-art system on PKU and MSRA benchmark datasets.

Acknowledgments. This work is supported by National Natural Science Foundation of China under Grant No. 61273318 and 60975054

¹ The results we got is slightly lower than the reported results in original paper. We have contacted the authors and they told us that the higher result they got was due to a bug in their code. Our results are considered to be reasonable with the bug free implementation

References

1. Emerson, T.: The second international Chinese word segmentation bakeoff. In: Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, vol. 133 (2005)
2. Duan, H., Sui, Z., Tian, Y., Li, W.: The cips-sighan clp 2012 Chinese word segmentation on microblog corpora bakeoff (2012)
3. Zhao, H., Kit, C.: An empirical comparison of goodness measures for unsupervised Chinese word segmentation with a unified framework. In: The Third International Joint Conference on Natural Language Processing (IJCNLP 2008), Hyderabad, India (2008)
4. Magistry, P., Sagot, B.: Unsupervised word segmentation: the case for mandarin Chinese. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, vol. 2, pp. 383–387. Association for Computational Linguistics (2012)
5. Wang, H., Zhu, J., Tang, S., Fan, X.: A new unsupervised approach to word segmentation. *Computational Linguistics* 37(3), 421–454 (2011)
6. Goldwater, S., Griffiths, T.L., Johnson, M.: A bayesian framework for word segmentation: Exploring the effects of context. *Cognition* 112(1), 21–54 (2009)
7. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. *Journal of the American Statistical Association* 101(476) (2006)
8. Casella, G., George, E.I.: Explaining the gibbs sampler. *The American Statistician* 46(3), 167–174 (1992)
9. Xu, J., Gao, J., Toutanova, K., Ney, H.: Bayesian semi-supervised Chinese word segmentation for statistical machine translation. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1, pp. 1017–1024. Association for Computational Linguistics (2008)
10. Kempe, A.: Experiments in unsupervised entropy-based corpus segmentation. In: Workshop of EACL in Computational Natural Language Learning, pp. 7–13 (1999)
11. Tanaka-Ishii, K.: Entropy as an indicator of context boundaries: An experiment using a web search engine. In: Dale, R., Wong, K.-F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS (LNAI), vol. 3651, pp. 93–105. Springer, Heidelberg (2005)
12. Jin, Z., Tanaka-Ishii, K.: Unsupervised segmentation of Chinese text by use of branching entropy. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, pp. 428–435. Association for Computational Linguistics (2006)
13. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1), 22–29 (1990)

Enhancing Chinese Word Segmentation with Character Clustering

Yijia Liu, Wanxiang Che, and Ting Liu

Research Center for Social Computing and Information Retrieval
School of Computer Science and Technology
Harbin Institute of Technology, China
{yjliu,car,tliu}@ir.hit.edu.cn

Abstract. In semi-supervised learning framework, clustering has been proved a helpful feature to improve system performance in NER and other NLP tasks. However, there hasn't been any work that employs clustering in word segmentation. In this paper, we proposed a new approach to compute clusters of characters and use these results to assist a character based Chinese word segmentation system. Contextual information is considered when we perform character clustering algorithm to address character ambiguity. Experiments show our character clusters result in performance improvement. Also, we compare our clusters features with widely used mutual information (MI). When two features integrated, further improvement is achieved.

Keywords: Brown clustering, Chinese word segmentation, semi-supervised learning.

1 Introduction

Chinese word segmentation is the first step of many NLP and IR tasks. Over the past years, word segmentation system's performance has been improved. However there are still some challenging problems. One of these problems is how to unearth helpful information from large scale unlabeled data and use this information to improve word segmentation system's performance. Former researchers have tried to use auto-segmented result of large scale unlabeled data[1] and statistical magnitudes like mutual information, accessory variety[2] to help the semi-supervised learning system. Performance improvement is achieved in their works.

In other tasks like NER, word clustering has been proved a helpful method to derive information from unlabeled data and improve the semi-supervised learning systems performance[3][4]. However, there hasn't been any work that applies clustering to word segmentation. The main reason is that there is no natural word boundary in Chinese. Traditional routine of clustering words cannot be applied to segmentation task directly. But, as character is the minimum unit of Chinese language, it's promising that we build clusters from character and use this character clustering information to assist word segmentation. In this

paper, we try to employ Brown clustering algorithm to build character-based clusters and embed contextual information into the character cluster. Finally we compile the clustering result into features and use this features to improve the word segmentation task. Experiments shows our character clustering results can help improving word segmentation performance.

The reminder of this paper is organized as follows. Section 2 describes the intuitive motivation and theoretical analysis of our character-based clustering method. Section 3 introduces the semi-supervised model we use to incorporate clustering results. Section 4 presents experimental results and empirical analysis. Section 5 gives some conclusion and future work.

2 Character-Based Brown Clustering

2.1 Brown Clustering

Data sparsity is always an issue in many NLP tasks. Capturing generality from unlabeled data is a promising way to address this issue.[4] Intuitively, character under the similar context environment tends to have similar function when compositing words. Supposing there is some criterion reflecting this similarity, we can use this criterion to help our word segmentation system. For example, in the following sentence “...期货中的做空行为...” (... the shorting in futures ...), “做空” (the shorting) is a financial term which barely occurs in newswire. While, similar context may occurs in “...管理中的违纪行为...” (... the disciplinary offence...) and this context is a typical newswire. This kind of similarity provides a clue for inferring the segmentation of “做空”.

In this paper, we view our clusters as *class-based bigram language model*. The *class-based bigram language model* considers the sentence as a sequence of characters and there is a hidden class behind each character in the sentence. Figure 1 illustrate the *class-based bigram language model* where c_i represent i_{th} character of the sentence and C_i is its cluster.

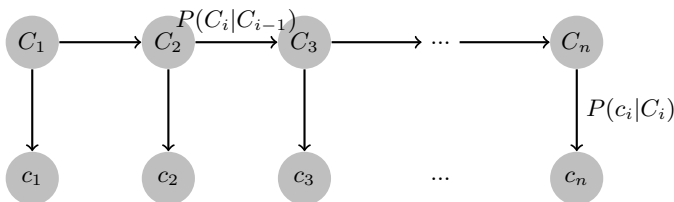


Fig. 1. Brown clustering model[4]

¹ This example occurs in the People Daily corpus.

Given a sentence $c_{1\dots n}$ consisting of n characters, probability of $P(c_{1\dots n})$ is modeled as follow:

$$\begin{aligned} P(c_{1\dots n}) &= P(c_1, c_2, \dots, c_n, C(c_1), C(c_2), \dots, C(c_n)) \\ &= \prod_{i=1}^n P(c_i|C(c_i))P(C(c_i)|C(c_{i-1})) \end{aligned} \quad (1)$$

To maximum the likelihood of the *class-based bigram language model*, we can derive a hierarchical clustering of words with a bottom-up clustering algorithm, which is known as Brown clustering algorithm[5][4]. The input of Brown clustering is a sequence of item. The output is a binary tree, which can be represented by a string of 01.

2.2 Unigram Character Clustering

In our unigram character clustering model, we follow our model definition as mentioned in section 2.1 and the cluster of a character only depends on the character itself. As result of the unigram character clustering model, each character is allocated with a single cluster. In our experiment, sentence is split into sequence of characters and brown clustering algorithm is employed on the sequence. Table 1 illustrates some result of the unigram character clustering model. For the clustering result shown in table 1, it seems our clustering model works well by clustering the Chinese digit into one cluster and Chinese metal name into another cluster. However, farther analytics will cast doubt on these results' effect. Syntactical and semantical function of same Chinese character varies under different circumstance. Simply dropping the contextual environment and clustering the character into mono-clustering will introduce a lot of ambiguity. Clustering result in Table 1 also shows this problem. The Chinese character “叶” can be a family name (translated as ‘Ye’), while it can also indicating part of the plant(translated as ‘leaf’). When used as a family name, “叶” is usually the leading character of a word. But when used as leaf, “叶” can composite word like “树叶”(leaf), “一叶障目”(a Chinese idiom which means having ones view of the important overshadowed by the trivial) and used as middle or end of a word. In following section, our experimental result also prove unigram character clustering doesn't work well.

Table 1. Clustering result, experiment are conducted on Gigawords setting number of clusters to 500

character cluster		character cluster		character cluster	
镓	0101111110	九	00111010111	张	00110010010
钢	0101111110	七	00111010111	王	00110010010
镁	0101111110	八	00111010111	李	00110010010
锂	0101111110	六	00111010111	叶	00110010010

2.3 Bigram and Trigram Character Clustering

To settle the problems mentioned above, same character under different context circumstance should be categorized into different clusters. We incorporate contextual information by considering character’s bigram and trigram. The model of $P(c_{1...n})$ changes into

$$\begin{aligned} P(c_{1...n}) &= P(c_1, c_2, \dots, c_n, C_1, C_2, \dots, C_n) \\ &= \prod_{i=1}^n P(c_i c_{i-1} | C(c_i c_{i-1})) P(C(c_i c_{i-1}) | C(c_{i-1} c_{i-2})) \end{aligned} \quad (2)$$

in bigram case and

$$P(c_{1...n}) = \prod_{i=1}^n P(c_{i+1} c_i c_{i-1} | C(c_{i+1} c_i c_{i-1})) P(C(c_{i+1} c_i c_{i-1}) | C(c_i c_{i-1} c_{i-2})) \quad (3)$$

in trigram case.

Table 2 shows our bigram character clustering result. First column of the result shows that our model cluster “叶” under the environment where it’s used as leading character and means leaf into same cluster. The second column capture the sentence segmentation like “...特级大师叶江川...”, “...设计师赵葆常...”, “...教师张百战...”². In this situation, the bigram “师叶” provide a clue for segmentation. The third column cluster the rare word “做空” into a cluster of common words. Analogously, trigram model gives similar results.

Table 2. Clustering result, experiment are conducted on Gigawords setting number of clusters to 500

character cluster		character cluster		character cluster	
叶脉	10001011	师赵	111010011	做空	1001101
叶片	10001011	师徐	111010011	选举	1001101
叶柄	10001011	师朱	111010011	遏制	1001101
叶子	10001011	师叶	111010011	抑制	1001101

3 Semi-supervised Learning Model

Previous study[1][2][6] has presented a simple yet effective semi-supervised method of incorporating information derived from large scale unlabeled data. Their method introduces new semi-supervised feature into robust machine learning model. In this paper, we follow their work and employ a conditional random fields (CRFs) model to incorporate character clustering results. This model is a character-based sequence labeling model, in which a character is labeled a tag representing the position of its position in word. We follow the work in [1] and select tagset of 6-tag style (B, B2, B3, I, E, S).

² All these three example is drawn from Chinese Gigawords(LDC2011T13).

3.1 Baseline Features

We employ a set of simple but widely used feature as baseline feature. The features we use are listed below.

- character unigram: c_s ($i - 2 \leq s \leq i + 2$)
- character bigram: $c_s c_{s+1}$ ($i - 2 \leq s \leq i + 1$), $c_s c_{s+2}$ ($i - 2 \leq s \leq i$)
- character trigram: $c_{s-1} c_s c_{s+1}$ ($s = i$)
- repetition of characters: is c_s equals c_{s+1} ($i - 1 \leq s \leq i$), is c_s equals c_{s+2} ($i - 2 \leq s \leq i$)
- character type: is c_i an *alphabet*, *digit*, *punctuation* or *others*

3.2 Mutual Information Features

In order to compare character clustering with traditional semi-supervised feature, we follow previous work[2] and feed mutual information to our semi-supervised model. Mutual information of two character is define as,

$$MI(c_i c_{i+1}) = \log \frac{p(c_i c_{i+1})}{p(c_i) p(c_{i+1})} \quad (4)$$

For each character c_i , we compute $MI(c_{i-1}, c_i)$ and $MI(c_i, c_{i+1})$ and round them down to integer. These integer value are integrated into CRF model as a type of features.

3.3 Clustering Features

We compile character clusters result into a kind of feature. When clustering algorithm is performed over large scale unlabeled data, a lexicon indicating ngram is cluster is maintained. For each character c_i in sentence, we extract the clusters of ngram and integrate them into our CRF model as a type of features.

For different clustering models, we extract different features. The clustering features are listed below,

- For our unigram character clustering model, brown clustering results of character in a window of 5 are extracted: $brown(c_s)(i - 2 \leq s \leq i + 2)$
- For our bigram character clustering model, we extract $brown(c_{i-1} c_i)$ and $brown(c_i c_{i+1})$
- For our trigram character clustering model, we extract $brown(c_{s-1} c_s c_{s+1})(i - 1 \leq s \leq i + 1)$

Here, $brown(x)$ represents the clusters of ngram x .

Table 3. Statistic of the corpus

Data set	# of sent.			# of words		
	train	test	dev	train	test	dev
CTB5.0	18,086	348	350	493,934	8,008	6,821
CTB6.0	23,417	2,769	2,077	641,329	81,578	59,947

4 Experiments

4.1 Settings

To test the character clustering’s effect on Chinese word segmentation, we select CTB5.0 and CTB6.0 as our labeled data. For these two data set, we split the data according to the recommendation in the document. Some statistic of the data is listed in Table 3.

Chinese Gigawords(LDC2011T13) is a remarkable achieve of unlabeled data, because of its huge quantity and broad coverage. Xinhua news(from 2000 to 2010) is chosen from Chinese Gigawords as unlabeled data in our experiment, which has about 500 million characters.

F1-score is used as measurement for our model. Define precision p as percentage of words that are correctly segmented in model output, and recall r as percentage of words that are correctly segmented in gold standard output. F1-score equals $\frac{2pr}{(p+r)}$.

We use a CRFs toolkit CRFSuite[7] to label sequential data. During the training parse, stochastic gradient descent is set as training algorithm. Two parameters *feature.possible_trainstions* = 1 and *feature.possible_states* = 1 is configed to enable negative features.

We use Liang’s implementation of Brown clustering algorithm³ to maintain the character clusters. Algorithm’s running time on an Xeon(R) 2.67GHz server is list in Table 4. We didn’t maintain clustering results of 500 and 1000 clusters in trigram case, because it would consume too much time.

Table 4. Brown clustering algorithm’s running time(hours) on Chinese Gigawords with different number of clusters. $c = x$ means number of clusters is x .

Model	$c = 100$	$c = 200$	$c = 500$	$c = 1000$
Bigram	3	11	73	245
Trigram	29	126	-	-

4.2 Results

According to previous study, number of clusters controls the representation capicity of clustering result[8]. We conduct experiment on various settings of

³ <https://github.com/percyliang/brown-cluster>

cluster number. Figure 2 shows our experimental result on CTB5.0 and CTB6.0. *baseline* means the CRFs model trained with baseline features. Symbol ‘+’ means model trained both with baseline features and the new features. Experiment results shows that in bigram case, model of $c = 500$ and $c = 1000$ respectively achieve best accuracy on CTB5.0 and CTB6.0’s development data. In trigram case, model of $c = 200$ perform better than that of $c = 100$ in both data set.

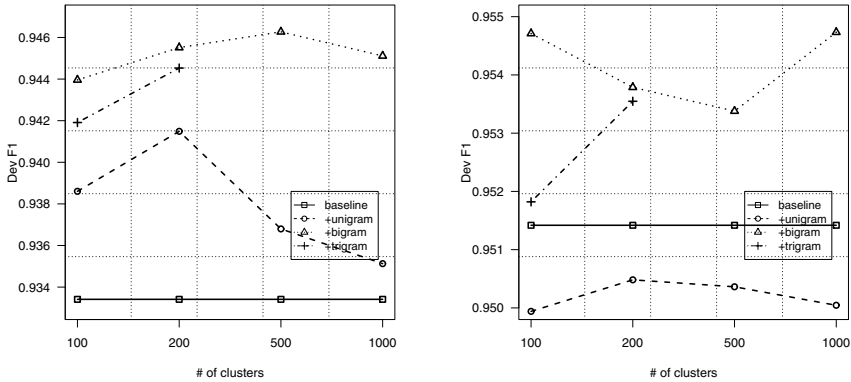


Fig. 2. Result on CTB5.0(left) and CTB6.0(right) development data

These results basically match what we expected in our former theoretical analysis. Unigram character clustering(+unigram) have almost no effect on Chinese word segmentation. However when increase the order of clustering model(+bigram,+trigram), increasement in F1-score is achieved. This result proves character clustering model considering contextual information is effective on Chinese word segmentation. Theoretically, trigram clustering exploits more contextual information and is expected to have better performance than bigram clustering. However, in both the CTB5.0 and CTB6.0, performance of model with trigram(+trigram) clusters is slightly lower than bigram model(+bigram). One reason maybe that trigram clustering introduce much noise due to exponential increased vocabulary size. Another reason for this may result from the limited number of clusters in trigram model. It takes more than 5 days to compute trigram brown clustering result of 200-clusters and it takes more of the 500-clusters cases.

In our bigram character clustering experiments, there is no significant improvement when we increase the number of clusters. But in trigram experiment, performance improvement is achieved when we transfer from 100-clusters to 200-clusters. Generally, we can conclude that fine-grained clusters help the word segmentation more.

After maintaining best c on development data, we conduct experiments on test data with best c configuration. Experiment result is show in Tabel 5. From this table, we can see our method outperform the baseline model. Significance tests

Table 5. Result of different models on CTB5.0 and CTB6.0 test data. bigram and trigram model is configed with best c in former experiments. $c = 500$ is set in the $+MI+bigram$ model.

Model	CTB5.0			CTB6.0		
	P	R	F	P	R	F
Baseline	0.9652	0.9733	0.9692	0.9478	0.9433	0.9455
+bigram	0.9699	0.9781	0.9740	0.9523	0.9504	0.9514
+trigram	0.9700	0.9760	0.9730	0.9506	0.9481	0.9494
+MI	0.9729	0.9804	0.9766	0.9530	0.9516	0.9523
+MI+bigram	0.9738	0.9808	0.9773	0.9533	0.9539	0.9536

between our method and baseline model also demonstrate that the improvements of our cluster features(+bigram,+trigram) is significant with $p - value < 10^{-4}$.

MI is a standard measurement of the association between character. Characters with stronger association is more likely to combine and composite a word. We compare our character clustering model that is configed with best c with model incorporated with MI. Table 5 shows the comparison results. In our experiment, we have seen that system incorporate with mutual information outperforms the character clustering model.

Although perform well, one limitation of MI is that it's computed only with local information between two characters. The character clustering which considering global information of a sentence makes a good complement of this limitation. In our experiments, we have seen further improvement on performance when two methods are combined.

5 Conclusion and Future Work

In this paper, we propose a method of building clusters from Chinese character. Contextual information is considered when we perform character clustering algorithm to address character ambiguity. Experimental result shows our character clustering result can help improve word segmentation performance.

In future, we will try to apply this method to some cross domain corpus. Also, we will try to use the character clusters to help other character-based NLP task like character-based Chinese parsing model.

Acknowledgments. This work was supported by National Natural Science Foundation of China (NSFC) via grant 61133012, the National “863” Major Projects via grant 2011AA01A207, and the National “863” Leading Technology Research Project via grant 2012AA011102.

References

1. Wang, Y., Kazama, J., Tsuruoka, Y., Chen, W., Zhang, Y., Torisawa, K.: Improving Chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In: Proceedings of the Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011 (2011)

2. Sun, W., Xu, J.: Enhancing chinese word segmentation using unlabeled data. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 970–979. Association for Computational Linguistics (2011)
3. Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discriminative training. In: Proceedings of HLT-NAACL, vol. 4. Citeseer (2004)
4. Liang, P.: Semi-supervised learning for natural language. PhD thesis, Massachusetts Institute of Technology (2005)
5. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. *Computational Linguistics* 18(4), 467–479 (1992)
6. Chen, W., Kazama, J., Uchimoto, K., Torisawa, K.: Improving dependency parsing with subtrees from auto-parsed data. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 2, pp. 570–579. Association for Computational Linguistics (2009)
7. Okazaki, N.: Crfsuite: a fast implementation of conditional random fields (crfs) (2007)
8. 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, pp. 384–394. Association for Computational Linguistics (2010)

Integrating Multi-source Bilingual Information for Chinese Word Segmentation in Statistical Machine Translation

Wei Chen, Wei Wei, Zhenbiao Chen, and Bo Xu

Interactive Digital Media Technology Research Center(IDMTech)
Institute of Automation, Chinese Academy of Sciences

Abstract. Chinese texts are written without spaces between the words, which is problematic for Chinese-English statistical machine translation (SMT). The most widely used approach in existing SMT systems is apply a fixed segmentations produced by the off-the-shelf Chinese word segmentation (CWS) systems to train the standard translation model. Such approach is sub-optimal and unsuitable for SMT systems. We propose a joint model to integrate the multi-source bilingual information to optimize the segmentations in SMT. We also propose an unsupervised algorithm to improve the quality of the joint model iteratively. Experiments show that our method improve both segmentation and translation performance in different data environment.

Keywords: Chinese segmentation, bilingual information, statistical machine translation.

1 Introduction

Different from most of the western languages, Chinese sentences are written without any spaces between the words. Word segmentation is therefore one of the most important steps of Chinese natural language processing tasks, such as statistical machine translation (SMT).

[1] showed that SMT system worked much better by segmenting the text into words than those treating each character as one “word”. While it is difficult to define what is a “correct” Chinese word segmentation (CWS), a generally accepted point is that the definition of “correct” segmentation should vary with different tasks. For example, Chinese information retrieval systems call for a segmentation that generates shorter words, while automatic speech recognition benefits from having longer words. However, it is difficult to define and poorly understood what is a satisfactory segmentation for SMT systems. [2] and [3] showed that the F-score, which is used generally to measure the performance of a segmentation on monolingual corpus, had nothing to do with the effect of the segmentation on SMT systems as a very high F-score may produce rather poor quality translations.

In spite of this, the common approach in most SMT systems has been to use an off-the-shelf monolingual CWS method. For instance, [4] proposed the N-gram generative language modeling based approach. [5] used the hierarchical hidden Markov Model

(HHMM) based method. [6] applied a sliding-window maximum entropy classifier to take CWS as a task of character tagging. Then [7] used Linear-chain conditional random fields (CRFs) [8] instead to take on the role of classifier and got a better result.

By the different existing methods, the fixed segmentations are applied in translation model training process even if they are sub-optimal and raise a series of problems as follows:

- Firstly, the specifications of monolingual CWS systems are not suitable for SMT. What’s more, the “best” specification in the bilingual corpus may differ from sentence to sentence (see Figure 1(a)), and it’s difficult to find it only through monolingual CWS method.
- Secondly, monolingual CWS methods often make a large number of mistakes on out-of-vocabulary(OOV) segmentation especially named entity(NE) segmentation (see Figure 1(b)).
- Thirdly, monolingual CWS methods are not good at dealing with the ambiguities in the Chinese text and will segment randomly because each segmentation is “right” (see Figure 1(c)).

Every problem can cause a chain mistake in the SMT system. Even so, it is poorly studied how to optimize the CWS in the machine translation system. [2] proposed two approaches to combine multiple word segmentations. [3] showed that neither character-level segmentation granularity nor Chinese-Treebank-style segmentation granularity

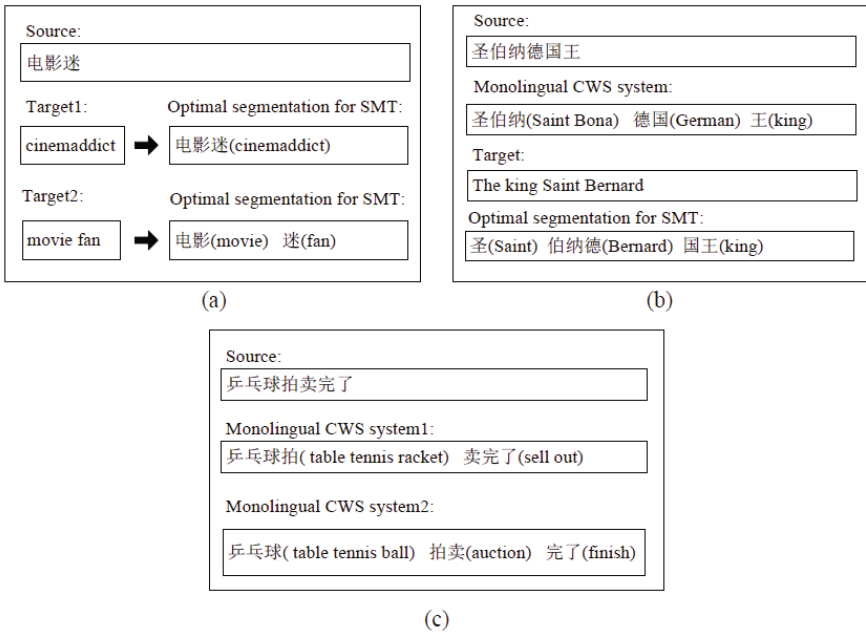


Fig. 1. Optimal segmentations in different situations using bilingual information

was suitable for SMT systems and it introduced a new feature to shorten the average word length produced by its CRF segmenter. However, the optimization in these papers is based on monolingual information and still keeps the problems above. [9] described a generative model which consisted of a unigram language model and an alignment model of both directions. Then it treated the word segmentation as a Hidden Markov Modeling problem of inserting and deleting spaces with the initial segmentations. But the approach suffers from the problems of local optimum because of the lack of linguistic specifications which introduces some mistaken alternatives. Furthermore, it couldn't address the issues of monolingual CWS systems only by the information of word alignment and called for multi-source information to be integrated.

To address the problems caused by monolingual CWS system and [9], we propose a joint translation model to integrate multi-source bilingual information into monolingual CWS methods to address the issues above. Firstly we apply a word-based translation model to rescore the alternative segmentations. We get the alternative set by the combination of CRF-based CWS system and N-gram language model based CWS system and rescore them by the way of cross-validation. Secondly, we take use of a phrase-based named entity (NE) transliteration model to integrate the information of bilingual NE into the model. Thirdly, we employ an English-Chinese dictionary and a Chinese synonym dictionary to make the model more accurate and effective. Finally, we propose an algorithm to improve the segmentation iteratively.

Our experiments show that the approach can generate a more satisfactory and correct segmentation for SMT systems and is very effective in improving the performance of machine translations.

2 Producing the Set of Alternative

2.1 Previous Work on Monolingual CWS

CRF-Based Model for CWS. CRF is an undirected graphical model trained to maximize a conditional probability [8] and is first used for CWS task by [10], which treats CWS task as a sequence tagging question. For instance, Chinese characters that begin a new word are given the START tag and Characters in the end of the words are given END tag. CRF-based model overcomes the problem of marking bias in generative models but has a shortage of prone to generate much longer word than other methods, which is harmful to SMT because it causes data sparseness.

N-gram Language Model for CWS. N-gram language model based method [4] treats CWS task as a hidden Markov modeling problem of inserting spaces into text. It defines two states between every pair of the characters of Chinese text: have a space or don't have a space between the pair of characters. N-gram language model has much weaker ability of recognizing OOV word than CRF-based model but it generates significant shorter words than CRF-based model, which meets our demand greatly.

2.2 Combination of CWS Systems

In order to produce the set of alternative effectively and accurately, we propose an approach to combine the two models above. First, we are given a Chinese sentence

$c_1^K = (c_1, c_2, \dots, c_K)$, where c_k indicates the character k in the sentence. Then, we get two segmentations by CWS models above: $f_{1CRF}^J = (f_1, f_2, \dots, f_J)$ produced by CRF-based model and $f_{1N-gram}^J = (f_1, f_2, \dots, f_J)$ produced by N-gram language model. In the sentence, we call a character as a word boundary when it is the ending (not the beginning) of a word in one of the segmentations. According to the description, we define four states of a character as follows:

- (1) C_{k+}^S indicates the character k is a word boundary both in f_{1CRF}^J and $f_{1N-gram}^J$.
- (2) C_{k-}^S indicates the character k is not a word boundary either in f_{1CRF}^J or $f_{1N-gram}^J$.
- (3) C_{k+}^D indicates the character k is a word boundary in f_{1CRF}^J while not a word boundary in $f_{1N-gram}^J$.
- (4) C_{k-}^D indicates the character k is a word boundary in $f_{1N-gram}^J$ while not a word boundary in f_{1CRF}^J .

Finally, we can describe our approach that produces the alternative segmentations as follows:

- (1) every C_{k-}^D between two adjacent C_{k+}^S in f_{1CRF}^J can be converted to C_{k-}^S or keep the original state in the sentence (see Figure 2(a)).
- (2) every C_{k+}^D between two adjacent C_{k+}^S in f_{1CRF}^J can be converted to C_{k+}^S or keep the original state in the sentence (see Figure 2(b)).

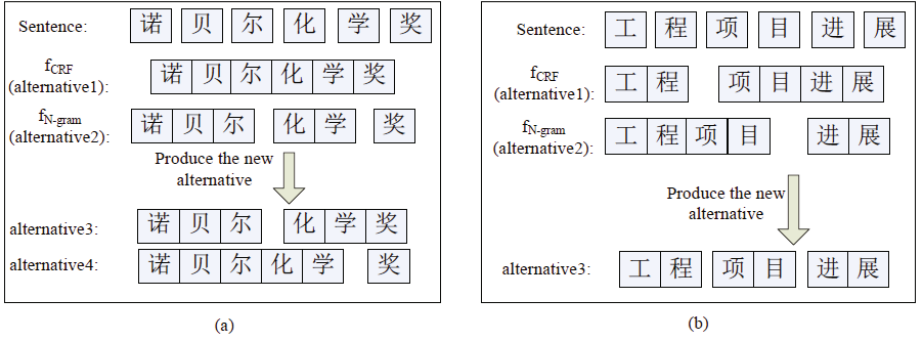


Fig. 2. Produce new alternatives in situation (1) and (2)

Then we can get the set of alternatives segmentations $set(f_1^J) = f_{1(1)}^J, f_{1(2)}^J, \dots, f_{1(L)}^J$ by combining each character's possible states and the set of alternatives in Figure 2(a) can be described as a graph (see Figure 3).

Each path that goes through the graph from left to right indicates an alternative segmentation and each alternative segmentation will be given a fixed value as their monolingual segmentation probability for the next process.

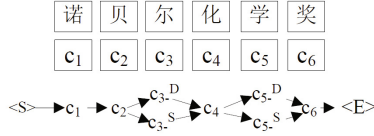


Fig. 3. The graph of the set of alternatives in Figure 2(a)

3 Joint Translation Model for Integrating Multi-source Bilingual Information

3.1 Word-Based Translation Model

For each parallel sentence (c_1^K, e_1^I) in the corpus, the observations are Chinese text c_1^K and English text e_1^I , and the hidden variable is the word segmentation f_1^J . In traditional SMT systems, we use monolingual CWS methods to select a “best” segmentation by assuming that the probability of the segmentation is conditional independent with the English text as follows:

$$f_1^J \text{ best} = \arg \max_{f_1^J} p(f_1^J | c_1^K, e_1^I) = \arg \max_{f_1^J} p(f_1^J | c_1^K)$$

however, it is proved by [9] that the assumption is harmful to the translation performance. Ignoring the assumption, we can select the “best” segmentation by the bilingual CWS probability as follows:

$$f_1^J \text{ best} = \arg \max_{f_1^J} p(f_1^J | c_1^K, e_1^I) = \arg \max_{f_1^J} \frac{p(e_1^I | f_1^J, c_1^K) * p(f_1^J, c_1^K)}{p(c_1^K) * p(e_1^I)}$$

where the c_1^K in $p(e_1^I | f_1^J, c_1^K)$ can be dropped because the c_1^K is fixed given the f_1^J .

It indicates the bilingual CWS probability of each alternative segmentation is determined both by the monolingual CWS probability $p(f_1^J, c_1^K)$ and the translation probability $p(e_1^I | f_1^J)$ and the probability $p(f_1^J, c_1^K)$ of each alternative segmentation are set to a fixed value as mentioned above, it can be ignored and the bilingual CWS probability thus is

$$p(f_1^J | c_1^K, e_1^I) \propto p(e_1^I | f_1^J)$$

for each alternative segmentation f_1^J , we compute the translation probability $p(e_1^I | f_1^J)$ with our word-based translation model. Considering the computing complexity, we take use of IBM model-1 in the process. As we can’t obtain the fixed alignment of each alternative, we take every possible alignment into account. Then the translation probability is derived by

$$p(e_1^I | f_1^J) = \sum_a P(e_1^I, a | f_1^J) = \frac{\varepsilon}{(J+1)^I} \prod_{i=1}^I \sum_{j=0}^J t(e_i | f_j) \quad (1)$$

where ε is the normalization factor to make all alternative segmentations' probability sum to one and "I" indicates the number of the words of English sentence e_1^I while "J" indicates the number of the words of the alternative segmentation f_1^J . $t(e_j|f_i)$ is the translation probability from Chinese word f_j to English word e_i which is given by our word-based translation model.

To avoid the problem of over-fitting, we introduce the thought of cross validation in the process of computing translation probability. That is, we compute the translation probability of sentence pair $(c_1^K, e_1^I)^i$ through the translation model which is trained on the corpus of the other sentence pairs without the current sentence pair $(c_1^K, e_1^I)^i$. Considering efficiency, we divide the corpus into two subsets and compute the probability of one using the translation model trained on the other subset.

3.2 English-Chinese Phrase-Based Named Entity Transliteration Model

As we mentioned in Section 1, it is really difficult for monolingual CWS methods to segment the proper names or technical terms which are defined as named entity (NE) correctly and suitably. As many different words can be the transliteration of the same English named entity since they pronounce in the same way, it causes a big problem of data sparseness, which can't be solved by the translation model in Section 3.1.

In this section, we propose a phrase-based named entity transliteration model to fill the gap.

Firstly, we get the transliteration model using an initial NE dictionary. We convert each named entity word pair (e_i, f_j) to a "sentence pair" (l_1^Y, c_1^X) by splitting e_i by letters and f_j by characters and train a standard English-Chinese phrase-based transliteration model using the open source translation system mooses.

Given the transliteration model and English word e_i , we convert the word e_i into an English "sentence" l_1^Y and derive the best transliteration of it as:

$$c_{1best}^X = \arg \max_{c_1^X} \prod_{x=1}^X \phi(\bar{c}_x | \bar{l}_x) d(start_x - end_{x-1} - 1) * \prod_{x=1}^{|e|} p_{LM}(c_x | c_1 \dots c_{x-1})$$

where $\phi(\bar{c}_x | \bar{l}_x)$ indicates the phrase translation probability of the phrase pair (\bar{c}_x, \bar{l}_x) . As an English named entity is generally transliterated from left to right, we don't need to reorder the translation and the value of reordering feature $d(start_x - end_{x-1} - 1)$ is fixed to $d(0)$. What's more, as we mentioned above, many different words pronounce in the same way. It doesn't matter which character is chosen and each will be a "correct" transliteration of the English word e_i . So the value of language model feature is set to a fixed value, too. Then the best transliteration of the "sentence" l_1^Y is derive as follows:

$$c_{1best}^X = \arg \max_{c_1^X} \prod_{x=1}^X \phi(\bar{c}_x | \bar{l}_x)$$

For each word pair (f_j, e_i) in the alternative segmentation and the corresponding parallel English sentence, we can integrate the feature of transliteration into the translation probability $p(e_1^I | f_1^J)$ in Formula (1). Then the probability that rescure the alternative segmentations is given by:

$$p(e_1^I | f_1^J) = \frac{\varepsilon}{(J+1)^I} \prod_{i=1}^I \sum_{j=0}^J [\lambda_1 t(e_i | f_j) + \lambda_2 f_{NE}(f_j, c_{1_{best}}^X)] \quad (2)$$

where λ_1 and λ_2 indicate the weights of word translation feature and named entity transliteration feature. The function of named entity transliteration feature $f_{NE}(f_j, c_{1_{best}}^X)$ is given by:

$$f_{NE}(f_j, c_{1_{best}}^X) = \begin{cases} 1 & \text{if the pinyin (pronunciation) of } f_j \text{ and } c_{1_{best}}^X \text{ is the same} \\ 0 & \text{if the pinyin (pronunciation) of } f_j \text{ and } c_{1_{best}}^X \text{ is different} \end{cases}$$

and the $c_{1_{best}}^X$ is given above.

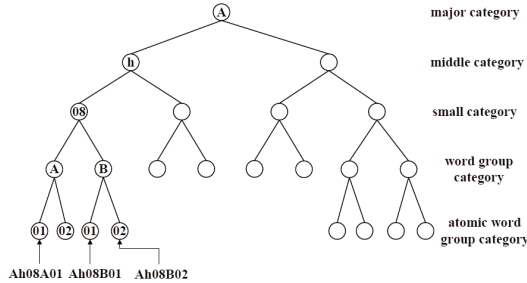
3.3 Integrating the Information of Dictionary

In order to promote the joint model to be more accurate, we put forward a dictionary-based model in this Section.

Firstly, we propose an English-Chinese translation dictionary $(e_i, T_i)^N$ where N indicates the number of items in the dictionary. Each item consists of an English word e_i and a set of the word's translations T_i .

However, it's impossible to collect all of the possible translations of English word e_i in the set T_i . What's more, it's common that replace the translation of English word with a synonym which may have a little difference in meaning with the English word.

To address the issue, we propose a dictionary of Chinese synonyms to compute the similarity of two Chinese words. The dictionary has five category's levels and every word is given one or more codes to indicate the categories of the word. The words given the same code have the almost same meaning. Based on the tree, we define the



semantic distance of two codes $SemDist(S_1, S_2)$ as the shortest distance from the point S_1 to point S_2 in the tree. For example, $SemDist(Ah08B01, Ah08B02) = 2$, $SemDist(Ah08B01, Ah08A01) = 4$. Then we define the similarity of two codes $SemSim(S_1, S_2)$ as follows:

$$SemSim(S_1, S_2) = \begin{cases} 1/SemDist(S_1, S_2) & \text{if } S_1 \neq S_2 \\ 0 & \text{if } S_1 = S_2 \end{cases}$$

The feature function of word pair (e_j, f_i) and the similarity of two words is therefore defined by

$$f_{DICT}(f_j, e_i) = \max_{\substack{W_1=f_j \\ W_2 \in T_i}} \begin{cases} 1 & \text{if } W_1 = W_2 \\ \max_{\substack{S_m \in \text{categoryOf}(W_1) \\ S_n \in \text{categoryOf}(W_2)}} \text{SemSim}(S_m, S_n) & \text{if } W_1 \neq W_2 \end{cases}$$

where the function *categoryOf*(W_1) return the set of codes of word W_1 . Finally, we extend the translation model described in Section 4.2 to

$$p(e_1^I | f_1^J) = \frac{\varepsilon}{(J+1)^I} \prod_{i=1}^I \sum_{j=0}^J [\lambda_1 t(e_i | f_j) + \lambda_2 f_{NE}(f_j, c_{1best}^X) + \lambda_3 f_{DICT}(f_j, e_i)] \quad (3)$$

where the λ_1 , λ_2 and λ_3 indicate the weights of word-based translation model, named entity transliteration model and dictionary-based model.

4 Iterative Algorithm

In this Section, as the algorithm showed in Algorithm 1, we present an iterative process to optimize the joint model and our segmentation in an unsupervised way.

In each iteration, we optimize the segmentation by the joint model, and then update the word-based translation model $M_{trans}(1), M_{trans}(2)$ and NE transliteration model M_{NE} by the optimized segmentations and the new NE dictionary D_{NE} .

The iteration will be stopped if the number of different segmentations of last iteration and current iteration is lower than a threshold h .

5 Experiment Setup

5.1 Data Set and Evaluation

We take the IWSLT machine translation task [11] for our experiment and our model is evaluated on the data track from two aspects: the segmentation performance on the training data set and the final translation performance on evaluation data set. The bilingual training corpus is a superset of corpora in the multi-domain collected from different sources including the training data of IWSLT task.

5.2 Baseline System and Translation System

We take CRF-based CWS method [7] as a baseline CWS method.

In order to highlight the translation performance, we use an out-of-the-box Moses¹ (2010-8-13 version). framework using GIZA++ [12] and minimum error rate training [13] to train and tune the feature weights of SMT systems. GIZA++ is used to get alignments from the bilingual training corpus with *grow-diag-final-and* option. The 4-gram LM is estimated by the SRILM toolkit [14] with interpolated modified Kneser-Ney discounting. We use the Moses decoder to produce all the system outputs, and score them with the BLEU-4 [15] score.

¹ <http://www.statmt.org/moses/index.php?n=Main.HomePage>

Algorithm 1. Iterative joint model training**Input:**

Bilingual corpus $(c_1^K, e_1^I)^n$, initial NE dictionary D_{NE} , English-Chinese dictionary D_{E2C} , Chinese synonyms dictionary D_{Syn}

Output:

- optimized segmented bilingual corpus $(f_{1opt}^J, e_1^I)^n$
- 1: divide the corpus into two subsets $(c_1^K, e_1^I)^{1..m}, (c_1^K, e_1^I)^{m+1..n}$
 - 2: get initial segmentations for each subset $(f_{CRF}, e_1^I)^{1..m}$ ($f_{N-gram}, e_1^I)^{1..m}$ and $(f_{CRF}, e_1^I)^{m+1..n}$ ($f_{N-gram}, e_1^I)^{m+1..n}$
 - 3: train initial word-based translation model $M_{trans}(1)$ for the first subset and $M_{trans}(2)$ for the other
 - 4: train initial NE transliteration model M_{NE} on D_{NE}
 - 5: get the set of alternative segmentations $set_i(f_1^J)$ for each sentence pair i $(c_1^K, e_1^I)_i$
 - 6: repeat
 - 7: $(f_{current}, e_1^I)^n \leftarrow (f_{opt}, e_1^I)^n, (f_{1opt}^J, e_1^I)^n \leftarrow \phi$
 - 8: **for** each sentence pair $(c_1^K, e_1^I)_i \in (c_1^K, e_1^I)^n$ **do**
 - 9: **for** each alternative segmentation $f_1^J \in set_i(f_1^J)$ **do**
 - 10: **for** each word pair (f_i, e_j) in the (f_1^J, e_1^I) **do**
 - 11: compute $t(e_j|f_i)$ by the M_{trans} that is trained on the other subset
 - 12: compute $f_{NE}(f_i, c_{1best}^I)$ by the M_{NE}
 - 13: compute $f_{DICT}(f_i, e_j)$ by the D_{E2C} and D_{Syn}
 - 14: add the word pair (f_i, e_j) to D_{NE} if $f_{NE}(f_i, c_{1best}^I) \neq 0$
 - 15: **end for**
 - 16: compute the score of f_1^J by the joint model
 - 17: **end for**
 - 18: select $f_{1best}^J \in set_i(f_1^J)$ with the highest score
 - 19: add (f_{1best}^J, e_1^I) to $(f_{1opt}^J, e_1^I)^n$
 - 20: **end for**
 - 21: retrain $M_{trans}(1), M_{trans}(2), M_{NE}$ by $(f_{1opt}^J, e_1^I)^n$ and D_{NE}
 - 22: until the number of different segmentations between $f_{current}$ and f_{opt} is lower than h
 - 23: **return** f_{opt}

6 Experiment

6.1 Segmentation Performance on Training Data Set

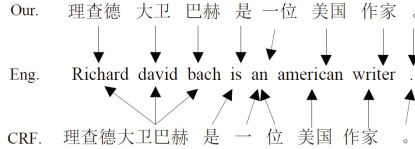
Firstly, we compare our model on the segmentation performance with currently widely-used monolingual CWS methods.

As we mentioned above, F-score can't measure the segmentations effectively in SMT systems and the CWS are related to SMT by a series of factors such as the specifications, OOVs, lexicons. None of these factors can be directly related to the SMT. Therefore, we compare our method with others in multiple factors on the training data as shown in Table 1. Considering computational complexity, we perform our method using only one iteration.

We can see that the number of running words generated by our method is close to the others. However, it produces a much smaller vocabulary than CRF and ICT [5] methods while keeps a high rate of unique words, which means that our method not only avoid

Table 1. Segmentation performance with different CWS methods on the training data

Method	Sents.	Tokens [M]	Voc. [K]	Unique Words[K]
ICT.	2M	18.80	214.1	41.0
CRF(base)		18.47	214.2	114.6
Our.		18.63	133.1	50.2

**Fig. 4.** Segmentations outputs with baseline and our method

data sparseness by shortening the common words, but also recognize the OOVs more accurate as the example shown in Figure 4.

6.2 Translation Performance on Task IWSLT

Then, we evaluate our method for word segmentation on the IWSLT machine translation task. The bilingual training corpus includes the training data of task and other corpus in the multi-domain collected from different sources. We take the open-source translation system mooses in the evaluation and use the evaluation corpus of (IWSLT 2005) [16] to optimize the model weights of mooses. Finally, we take the evaluation corpus of (IWSLT 2007) [11] to evaluate the translation performance.

For a fair comparison, we evaluate on various CWS methods including ICTCLAS [5], CRF-based method [7], N-gram language model based method [4], GS [9] and our method as shown in Table 2.

Furthermore, we replace the monolingual CWS methods, i.e., CRF-based method and N-gram language model based method, with another two monolingual methods, and then integrate parts of our joint model or our full model into them to evaluate the translation performance using the same evaluation corpus as above. The results are shown in Table 3.

Table 2. Translation performance with different CWS methods on IWSLT 2007[% BLEU]

Method	ICT.	CRF(base)	N-gram	GS	Our.
BLEU	39.62	39.25	38.22	39.99	41.23

It can be seen that each part of our joint model can improve the translation performance effectively. It also can be found that even if ICT system has a better translation performance than N-gram, it obvious that N-gram method are more adaptive to combine with CRF method using the joint model because N-gram is prone to segment the OOVs into characters and thus is fit for our method.

Table 3. Translation performance with integrating the joint model to another CWS methods[% BLEU]: A = word-based translation model, B = phrase-based NE transliteration model, C = dictionary-based model

Joint model	ICT.	CRF	N-gram	CRF
monolingual	39.62	39.25	38.22	39.25
+A	40.26		40.45	
+A+B	40.62		40.96	
+A+B+C	40.94		41.23	

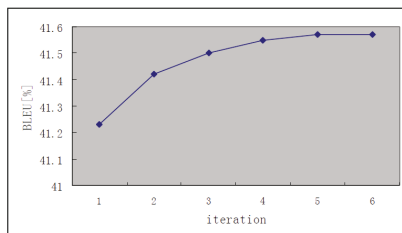


Fig. 5. Translation performance with the joint model of each iteration[% BLEU]

As our joint model can be optimized iteratively, we use 6 iterations (using N-gram and CRF methods) over the training corpus and evaluate the translation performance for each iteration as shown in Figure 5. We get the final BLEU at iteration 6 in Figure 5 is 41.58.

We compare the translation outputs using our method with the baseline method and list two examples in Table 4.

Table 4. Translation outputs with baseline and our methods

	Example1	Example2
Eval	咖啡还没有上来。	我也是啊。他们真的很棒啊。
Base	not coffee .	they also is . I really wonderful .
Our.	coffee hasn't come yet .	me too . they are really wonderful .
REF	my coffee hasn't come yet .	me , too . They play really well .

7 Conclusion and Future Work

In this paper, we showed that it is effective to improve the performance of SMT system by introducing multi-source bilingual information to CWS system. We proposed a joint model and an iterative algorithm and our experiments showed that our method outperformed the other CWS approaches in terms of not only the word segmentation performance but also the translation quality. It is also proved that each sub-model of our joint model is effective and the iterative algorithm works well. In future work, we plan to make our joint model more accurate to select the segmentation for SMT system better.

References

- [1] Xu, J., Zens, R., Ney, H.: Do we need Chinese word segmentation for statistical machine translation. In: Proc. of the Third SIGHAN Workshop on Chinese Language Learning, Barcelona, Spain (2004)
- [2] Zhang, R., Yasuda, K., Sumita, E.: Improved Statistical Machine Translation by Multiple Chinese Word Segmentation. In: Proceedings of the Third Workshop on Statistical Machine Translation, pp. 216–223 (2008)
- [3] Chang, P.-C., Galley, M., Manning, C.D.: Optimizing Chinese Word Segmentation for Machine Translation Performance. In: Proceedings of the Third Workshop on Statistical Machine Translation, pp. 224–232 (2008)
- [4] Teahan, W.J., Wen, Y., McNab, R., Witten, I.H.: A Compression-based Algorithm for Chinese Word Segmentation. *Computational Linguistics* 26(3), 375–393 (2000)
- [5] Zhang, H.-P., Yu, H.-K., Xiong, D.-Y., Liu, Q.: HHMM-based Chinese lexical analyzer ICTCLAS. In: Proceedings of the Second SIGHAN Workshop on Chinese Language Learning, pp. 184–187 (2003)
- [6] Xue, N.: Chinese Word Segmentation as Character Tagging. *Computational Linguistics and Chinese Language Processing* 8(1), 29–48 (2003)
- [7] Tseng, H., Chang, P., Andrew, G., Jurafsky, D., Manning, C.D.: A conditional random field word segmenter for Sighan bakeoff 2005. In: Proc. of the Fourth SIGHAN Workshop on Chinese Language Processing (2005)
- [8] Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning (2001)
- [9] Xu, J., Gao, J., Toutanova, K., Ney, H.: Bayesian Semi-Supervised Chinese Word Segmentation for Statistical Machine Translation. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 1017–1024 (2008)
- [10] Peng, F., Feng, F., McCallum, A.: Chinese segmentation and new word detection using conditional random fields. In: Proceedings of the 20th International Conference on Computational Linguistics, p. 562 (2004)
- [11] IWSLT: International workshop on spoken language translation home page (2007), <http://www.slt.atr.jp/IWSLT2007>
- [12] Och, F.J., Ney, H.: Improved statistical alignment models. In: Proceedings of ACL, pp. 440–447 (2000)
- [13] Och, F.J.: Minimum error rate training in statistical machine translation. In: Proceedings of ACL, pp. 160–167 (2003)
- [14] Stolcke, A.: SRILM - An extensible language modeling toolkit. In: Proceedings of ICSLP, pp. 901–904 (2002)
- [15] Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: A method for automatic evaluation of machine translation. In: Proceedings of ACL, pp. 311–318 (2002)
- [16] IWSLT: International workshop on spoken language translation home page (2005), <http://www.slt.atr.jp/IWSLT2005>

Interactive Question Answering Based on FAQ

Song Liu^{1,2}, Yi-Xin Zhong¹, and Fu-Ji Ren²

¹ School of Computer Science, Beijing University of Posts and Telecommunications,
10th Xitucheng Road, Beijing 100876, China

² Faculty of Engineering, The University of Tokushima, 2-1 Minamijosanjima,
Tokushima 770-8506, Japan

Songliu84@gmail.com, yxzhang@ieee.org, ren@is.tokushima-u.ac.jp

Abstract. A question answering system receives the user's question in nature language, and answers it in a concise and accurate way. An interactive question answering (IQA) provides a natural way for users to express their information requirement. There are two key points for IQA. The first is how to answer a user's question in a continuous question answering process. The second is the way that the question answering system interacts with the user. In this work the answers are from FAQ knowledge base which is extracted from community question answering web portals. The syntactic, semantic and pragmatic features between question and candidate answers and context information are used to construct models by ranking learning method to extract the answers. And the question answering system requests user to feedback of the answer. It is a naive and effective interactive method. The results of experiments show that our method is effective for interactive question answering.

Keywords: context, FAQ, Interactive question answering, ranking learning.

1 Introduction

A question answering system receives the user's question in nature language, and answers it in a concise, accurate and natural way. Question answering system analyzes the question to acquire the information requirement of user. And then based on the knowledge question answering system convert the information requirement into constraint condition while searching answers in knowledge space. The interaction between user and question answering system is introduced in the interactive question answering. There are two differences between interactive question answering and single round question answering. The first difference is that interactive question answering is a continuous question answering. How to use the context information is one key point. The second difference is that the question answering system interacts with users besides answering questions. It is the second key point of interactive question answering.

In this work, the answers are from the FAQ. FAQ provides information from the question-answer pair. The community question answering in web portals brings out abundant FAQs. In this work the FAQ is from the Baiduzhidao. For the first key point

of interactive question answering, the ranking learning method is used to train a statistic model to predict answers. The features that describe the training and testing instances are from the question, related FAQ and question answering context. And the syntactic, semantic and pragmatic information is concerned to extract the features.

For the second key point, the interaction between the question answering system and user is designed as that after answering a user's question the system will request the user to feedback whether he is satisfied with the answer. This interactive mode has two advantages. First is that the white or black feedback is easy to be caught and fully used by QA system. Secondly this kind of feedback matches FAQ. If the feedback is positive the question and answer will be added into the FQA base. If the feedback is negative, QA system will provide another answer based on the question and repudiated answer.

The rest of this paper is organized as follows: Section 2 introduces related work. Section 3 is about the model and syntactic, semantic and pragmatic features in context question answering. Section 4 describes the interaction between user and QA system. Section 5 is the experiments and results. Section 6 is conclusions.

2 Related Work

There are two key points for interactive question answering. The first is how to get answers in continuous question answering. The second is the interaction between QA system and users. To the first point, since 2004 in TREC QA task [1], a group of questions were around one topic. Hence a question was related to the question answering context. In real interaction the topic transformation is possible. The questions may not be around one topic. Yang estimated whether the topic is transferred through anaphora and ellipsis by decision tree [2]. Sun used center theory to deal with the anaphora in context question answering [3]. Sun [4] and Chai [5] used discourse theory in context question answering. Kirschner adopted logistic regression with contextual information to find answers in context QA [6]. There are three methods to obtain the data of context question answering. The first was getting questions from the QA evaluation such as TREC QA task [3]. However there was a gap between this kind of data and the realistic context question answering. The second method is wizard of OZ [7] which simulated the interaction between user and QA system to get the question answering data. The third method was collecting the question answering data while using the question answering system. This kind of data is the most realistic data. However, this kind of data is limited by the ability of the QA system.

The second key point is the method of interaction between user and QA system. The TREC QA task also explored the interaction between QA system and user. Interactive QA was first introduced in TREC 2006[9]. And in TREC 2007 QA task[10] reviewers interact with the QA system online. As reported, for most systems the answers are improved after interaction. However the improvements were not significant. Hickl employed CRF model to construct question-answer pairs. And then it showed the question answering pair to user to impact the following question that users will ask in

the next round [11]. Misu used reinforcement learning to learn the interacting strategy in the interactive process [12]. Similarly, adaptive learning was also used in an interactive question answering [13].

The FAQ is a kind of knowledge source that is easy to use by QA system. In 1997 Burke researched the FAQ question answering in semantic level [14]. Kolkata discussed how to get answers when there are no related questions in question-answer pairs [15]. Cong used sequence pattern based classification method to extract FAQ pairs form forum and a community question answering [16]. In this work ranking learning method is used to select answers [17]. Ranking learning methods are widely used in information retrieval [18] and question answering [19].

3 Context Question Answering in Interactive Question Answering

This section introduces the first key point for interactive question answering, how to extract answers in context question answering. The answer extraction is converted to a ranking learning process. It is supervised learning that uses labeled instances to train the statistic model. In the following part of this section the support vector based ranking learning methods and the syntactic, semantic and pragmatic features in context question answering are introduced.

3.1 Support Vector Based Ranking Learning Method

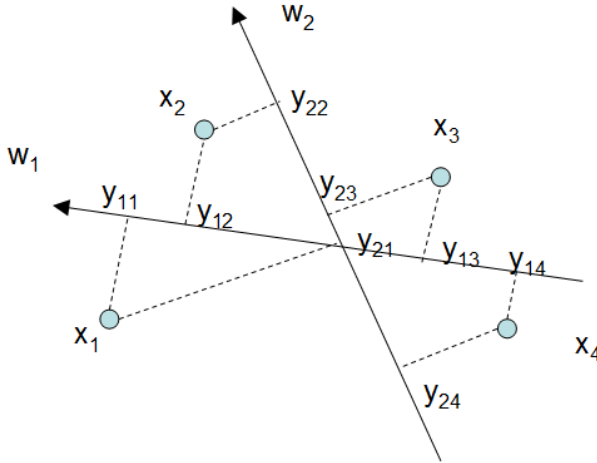
Two support vector based ranking learning methods Ranking SVM[20] and SVM-MAP[21] are used to rank the candidates question-answering pairs. The idea of support vector was first exploited in support vector machine (SVM) [22]. The core of SVM is finding the plane (support vector) $\hat{y} = y_i(w \cdot x_i + b)$, which makes the training data correctly classified and the geometric intervals maximize. The optimization of SVM is described in the following equations. ξ_i is the slack variable for linearly inseparable conditions. The parameter C is the penalty parameter.

$$\min_{w,b,\xi} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (1)$$

$$s.t. \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N \quad (2)$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, N \quad (3)$$

Support vector based ranking learning has some differences from SVM classification. The mainly difference is that output Space of ranking learning is an ordered sequence space. The optimization target of support vector based ranking learning is that the order of the output sequence is correct and maximizes the interval of distances between the elements which are mapped on the plane.



A group of instances (x_1, x_2, x_3, x_4) in the input space X . The correct ordering is $\langle y_1, y_2, y_3, y_4 \rangle$. After mapping the instances onto plane w_1 the ordering of them is $\langle y_{11}, y_{12}, y_{13}, y_{14} \rangle$. And After mapping the instances onto plane w_2 the ordering of them is $\langle y_{22}, y_{23}, y_{21}, y_{24} \rangle$. Hence the plane w_1 can order the instances correctly.

Fig. 1. The input instances are mapped onto different plane

Two support vector based ranking learning methods will be introduced. The first is Ranking SVM which is a pair-wise approach. The second is SVM-MAP which is a list-wise approach. The mainly difference between the two approaches is the difference between their lose/risk functions.

In Ranking SVN the risk function is described by Kendall's τ [23]. Kendall's τ is a metric to measure the consistency of two finite strict orderings. For two finite strict orderings r_a and r_b which are in the same space, $r_a \subset D \times D$ and $r_b \subset D \times D$, the Kendall's τ is defined as:

$$\tau(r_a, r_b) = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{\binom{m}{2}} \tag{4}$$

In the two orderings. If a pair $d_i \neq d_j$ has the same order in r_a and r_b , the pair is concordant. Otherwise the pair is discordant. P is the number of the concordant pairs and Q is the number of the discordant pairs. And m is the number of the elements in ordering. The summation of P and Q is $\binom{m}{2}$. The higher of Kendall's τ , the more consistency the two orderings are.

SVM-MAP is a support vector based list wise ranking learning approach. The risk function of SVM-MAP is described by mean average precision (MAP) [24]. MAP is also a metric which measure the consistency of two finite strict orderings. It is widely used in the evaluation of information retrieval and question answering. Average

precision is the basis of MAP, which is calculated as equation 9. In the equation $p(i)$ is the precision of the elements from the first to the i th. And $\Delta r(i)$ is the variation of recall in the i th position. It is the difference between $r(i-1)$ and $r(i)$. And $r(i)$ is the precision of the elements from the first to the i th. On the right of equation 9 $rel(i)$ represents whether the i th element is a correct answer for the question. Average precision describes the consistency for the two orderings from precision and recall. MAP is the mean of APs for several groups of orderings.

$$AP = \sum_{i=1}^n p(i)\Delta r(i) = \frac{\sum_{i=1}^n (p(i) \times rel(i))}{\text{element number}} \quad (5)$$

$$MAP = \frac{1}{qnum} \sum_{k=1}^{qnum} AP = \frac{1}{qnum} \sum_{k=1}^{qnum} \left(\sum_{i=1}^n p(i)\Delta r(i) \right) \quad (6)$$

The two support vector based ranking learning methods, Ranking SVM and SVM-MAP has been introduced. The mainly differences between them are the definitions of the risk function. And the two ranking learning approaches will tested in the experiments.

3.2 The Features for Context Question Answering

The features to describe the instances of context question answering from syntactic, semantic and pragmatic level. Here the syntactic, semantic and pragmatic from question Q and candidate question-answer(Q', A') pairs are first introduced.

3.2.1 Syntactic Feature

The syntactic feature describes the similarity between question and candidate question-answer pair in grammatical form. The overlap of words are used to calculate the syntactic similarity[19]. The overlap of words is the proportion of the concurrence words in the sentence. The overlap of words between question Q and the question Q' and answer A' of candidate question-answer pair(Q', A') are calculated separately. Firstly the question Q and QA pair (Q', A') are segmented and tagged the POS and the entities. Then the verbs, nouns, adjectives and entities are retained to calculate the overlap of words between question Q and question Q' and answer A' in candidate QA pair, as the equation 7 and 8. C is the number of the overlap words in sentence. And n is the number of words of a sentence.

$$\text{overlap}(Q, Q') = \frac{C_{QQ'} + C_{Q'Q}}{n_Q + n_{Q'}} \quad (7)$$

$$\text{overlap}(Q, A') = \frac{C_{QA'} + C_{A'Q}}{n_Q + n_{A'}} \quad (8)$$

3.2.2 Semantic Features

The semantic features indicate the similarity between question Q and candidate QA pairs (Q', A') in content. The semantic similarity between two sentences is calculated

based on the word semantic similarity. Here the word semantic similarity is calculated based on Howent [25]. We refer Liu's method[26] to calculate the word semantic similarity between words based on sememes. The similarity between two sememes is calculated based on the distance of the two sememes in the sememes tree(equation 9).

$$Sims(p_1, p_2) = 1 - \frac{dis(p_1, p_2)}{d_1 + d_2} \quad (9)$$

And the similarity between concepts is calculated based on the sememes similarity. There are four kinds of sememes describing a concept. They are first sememe, basic sememe, relational symbol and relational sememe. The similarities of the four kinds of sememes between two concepts are calculated separately. And then the semantic similarity is calculated as equation 10, where $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$, $\beta_1 \geq \beta_2 \geq \beta_3 \geq \beta_4$.

$$Simw(S_1, S_2) = \sum_{i=1}^4 \beta_i \prod_{j=1}^i Sims_j(S_1, S_2) \quad (10)$$

The semantic similarity between sentences is calculated based on the word semantic similarity. Equation 20 shows how to calculate the sentence semantic similarity. The sentence similarities between question Q and question Q and answer A in the candidate QA pair are calculated as equation 12 and 13.

$$simsem(q, s) = \frac{\sum_{i=1}^n \max_{j=1, \dots, m} (sim(qnoun_i, snoun_j)) + \sum_{i=1}^p \max_{j=1, \dots, q} (sim(qverb_i, sverb_j))}{num(nouns) + num(verbs)} \quad (11)$$

$$simsem(Q, Q') = \frac{\sum_{i=1}^n \max_{j=1, \dots, m} (sim(Qnoun_i, Q'noun_j)) + \sum_{i=1}^p \max_{j=1, \dots, q} (sim(Qverb_i, Q'verb_j))}{num(nouns) + num(verbs)} \quad (12)$$

$$simsem(Q, A') = \frac{\sum_{i=1}^n \max_{j=1, \dots, m} (sim(Qnoun_i, A'noun_j)) + \sum_{i=1}^p \max_{j=1, \dots, q} (sim(Qverb_i, A'verb_j))}{num(nouns) + num(verbs)} \quad (13)$$

3.2.3 Pragmatic Features

Pragmatic feature describes the effect of information for the goal of subject. In question answering system the subject is the user. And the goal of the user is obtaining the answer that satisfies the user's information requirements. In question answering the pragmatic information is the information that indicates the whether the answer can meet the user's information requirement. In this work, the pragmatic information is from the question Q and Q' in candidate QA pair. The pragmatic feature of question Q describes the expected speech act of answers. The user has expectation on the speech act of answers. For example, when a user asks "Why France rejected the EU constitution treaty?", the expected speech act of answer is "explanation". From the expected act of answer, questions are classified in five categories: statement, instruct, explanation, verifying and opining.

Table 1. Expected answering acts

Expected act	Description	Question
statement	Information, claim, or announcement	When will the next train arrive? What are greenhouse gases? Who is Justin Bieber?
instruct	An idea or a manner that is suggested	How can I drive from Beijing to Shanghai?
explanation	An explanation	Why the American invasion of Iraq?
Verifying	An affirmation or negation	Were you born in 1990?
Opining	An opinion, evaluation, or attitude	How about my new longuette?

Table 2. Common Chinese interrogatives

types	interrogatives
interrogative pronoun	什么(what), 谁(who), 哪里(where), 哪儿(where), 何(what), 孰(which), 哪个(which), 哪(where), 啥(what), 哪些(which)
interrogative adverbs	为何(why), 怎样(how), 怎么办(how to), 为什么(why), 咋(how, why), 怎(how), 多少(how much), 多高(how tall), 多久(how long), 多长(how long), 多重(how heavy), 何如(how), 怎么(how), 为啥(why), 怎么样(how), 如何(how), 何以(why), 缘何(why)
Confirmation verb	*不* (* not *), *没* (* not *), *否* (* not *), *(了)没* (* not)
modal particle	吗(ma), 呢(ne), 么(me) ¹

A maximum entropy [27] method is used to classify the expected speech acts of answers. The classification features include n-gram, interrogative, the words modified by interrogative and syntactic structure. We collect 3124 questions by a search engine based on Chinese interrogatives and label the expected speech act of answer. And 70% data is used to train the model and 30% data is used to test it. The results are shown in table 3. And the average F score is 91.3%

Table 3. Evaluation of classification for expected questioning acts

	Statement	Instruct	Explanation	Opining	Verifying
Precision	0.933	0.852	0.961	0.766	0.769
Recall	0.976	0.784	0.805	0.855	0.682
F-score	0.954	0.790	0.876	0.808	0.723

And for the question Q' in the candidate QA pair, the expected speech act is also classified as another pragmatic feature. And if the two expected speech acts are matched, there is more possibility that the candidate QA pair is the answer for the question.

3.3 Context Features

In context question answering, the previous questions and answers construct the context. The context features include whether the topic of QA is continuous and the syntactic and semantic similarities between the candidate QA pair and context.

¹ They are the common Chinese modal particles. We labeled them with pinyin.

The transferring of topic in continuous question answering is common. If the QA topic transfers, the question has no relations with the context, and the context information cannot assist to find answers. Hence whether the QA topic is continuous is an important feature when using the context information.

Here whether the topic continuity is converted to a dichotomy problem. SVM is an effective classification method in dichotomy problem. Here the tool libsvm[28] is used. RBF kernel is chosen and the parameter c and r are confirmed by the tool grip. And the features for classification include the features from the current question and the features from context. Firstly the features from the question are introduced. The anaphora is an important feature. If the question contains anaphora, it is high possibility the question has relation with context. And some conjunctions and adverbs such as “既然” (since) and “那么” (then) also show the continuous relation. Ellipsis is also an important feature. We use the dependency of the question to judge the ellipsis in question. If the subject is lacking in the question, the ellipsis are confirmed. The second kind of features is from the question and context. The syntactic and semantic similarities between question and question answer pair in the previous round are introduced as features.

The training data have two sources. First is the IT question answering in CSDN community. Second is the Confucius and analects of Confucius question answering in Baiduzhidao. We collect 400 group context question answering and tag the continuity manually. And 5 times cross validation is used. Table 4 shows the result.

Table 4. The result of context continuity

	precision	recall	F score
continuous	0.714	0.781	0.746
discontinuous	0.829	0.773	0.8

The features of the second kind are the context syntactic and semantic similarity features. These features are: (1) the syntactic and semantic similarities between the question Q' in candidate QA pairs and the last user question Q_p , (2) the syntactic and semantic similarities between the question Q' in candidate QA pairs and the last answer A_p , (3) the syntactic and semantic similarities between the answer A' in candidate QA pairs and the last user question Q_p , (4) the syntactic and semantic similarities between the answer A' in candidate QA pairs and the last answer A_p .

4 The Interaction between QA System and User

The second key point of interactive question answering is the interaction mode. As presented by the TREC CIQA task, complex interaction did not assist to get answers signally. In this work a naive and effective interaction mode is adopted. The interactive mode is that after providing answers, the question answering system will ask user for feedback whether he is satisfied with the answer. It is the direct representation of the answer effect. In this interaction mode the form of user's feedback is restricted. Hence the feedback information is easy to be obtained and used.

There are positive and negative feedbacks from the user. For the positive feedbacks, user's question and system's answer are combined together as the QA pair and is stored in the FAQ knowledge base. It makes the knowledge of QA system increases in the process that the user uses a question answering system. For the negative feedbacks, QA system should supply a new answer for the user. When finding the new answer, the information from question and previous answer is used. The syntactic and semantic similarity between question Q and new answer are calculated. And the semantic similarity negative answer and new answer are also calculated. Then the score for new answer is calculated as equation 14. It follows the hypothesis that the more related the new with the question and the less related to the negative answer, the more possibility the new answer is correct.

$$Score(Q, A^-, A) = \frac{syn(Q, A) + sem(Q, A)}{syn(A, A^-) + sem(A, A^-)} \quad (14)$$

5 Experiment

First the method to acquire the data in question answering is introduced. In this paper the QA system is FAQ based QA system about the Analects of Confucius. The FAQ is from Baiduzhidao. Baiduzhidao is a portals community of question answering. In community question answering the answers are also from users which contain the domain experts. And users can vote, commit the answers to filter the best answers. Hence the community question answering is an effective source to get the FAQs. Here we crawled more than 26000 questions about the the Analects of Confucius. And about 7100 QA pairs that are labeled "best answer" are restored in the FAQ knowledge base.

Baiduizhiao is also used to acquire the training and testing data. Volunteers are required to supply 10 questions about the Analects of Confucius. volunteers can get the answers from Baiduzhidao, and consider the next question based on the answer. Form 10 volunteers 100 questions are collected. And after filtering the improper questions 10 groups 90 questions are retained. This method simulates the interaction process between users and QA system, so that it is a Wizard of oz method.

The 10 groups questions are divided into 5 parts. And 5 cross validation is used to evaluate the results. The evaluation metrics are MAP and p@1. MAP has introduced in section 3.1. P@1 is the precision of the first answer. In this experiment the effect of context features is evaluated. The results with and without context features are compared.

The results show that the results of SVM MAP are better than Rank SVM. And it is consistent with the related works [21]. And after adding the context features the MAP and p@1 using SVM MAP are both improved. And for Ranking SVM the MAP decreases slightly. These prove that the context features are important for continuous question answering. And after feature selection the MAP and p@1 are both increase for SVM MAP and Ranking SVM. And the results are also comparable with recent related work [15].

Table 5. The results for continuous question answering

feature	Ranking SVM		SVM MAP	
	map	p@1	map	p@1
Without context features	0.635	0.533	0.737	0.644
With context features	0.631	0.522	0.769	0.677

The second part of the experiments is about the interaction method between QA system and user to verify the effect of the interaction method. Based on the previous experiments, there are 62 answers are correct for 90 questions. For the correct answers users give positive feedbacks. These question answer pairs are stored in the FAQ knowledge base. And for the rest 28 answers with negative feedback, the method in section 4 is used to find a new answer for the user. And 15 new answers are correct. The precision of the second round answers is 0.536. And combining the first round answers and second round answers the total precision is 0.856. It proves that the interaction between QA system and user is effective and helpful to find the collect answer.

6 Conclusions

By interactive question answering user can obtain information conveniently. The interactive answering faces two major problems. The first is how to answer the user question in the process of continuous answering, which is mainly manifested as how to use context information. The Second is the interaction mode between QA system and the user. For the continuous question answering, this paper adopts the frequently answers to questions (FAQ) as the source of the answer and uses ranking learning methods based on support vector to build model. The features of describing training and test instances mainly come from two sources: one is the syntax, semantic and pragmatic features of the candidates question answer pairs, the other is the continuity of the question answering and the syntax and semantic features from context. As to the interaction mode between QA system and the user, the QA system asks users whether they are satisfied with the answer. This interaction is simple and effective, because the information obtained is easily understood and used by the QA system. It can provide the basis for a QA system to get new answers or adding correct answers to the questions to the knowledge base. Experiments show that it is effective to answer user questions by using the ranking learning method and multiple features. And the interaction between QA system and users further significantly improve the accuracy of the QA system to answer user questions. Future studies need to address the following questions. Firstly more features describing the relations between questions and answers are needed to understand the questions and answers and to improve the performance of QA system. The second is the interaction way between QA system and the user. A naive interactive way is used in this paper, by this way, the QA system is easy to understand and use the information. However, the expression of the user's information is limited, so in future studies it should also be concerned about the way that enables users to more freely express their intention.

References

1. Voorhees, E.M.: Overview of the TREC 2004 robust retrieval track. In: Proceedings of TREC 2004 (2004)
2. Yang, F., et al.: A data driven approach to relevancy recognition for contextual question answering. In: Proc. of the IQA Workshop at HLT-NAACL 2006, pp. 33–40. Association for Computational Linguistics, New York (2006)
3. Son, M., Chai, J.: Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems. Special Issues on Intelligent User Interfaces* 20(6), 511–526 (2007)
4. Sun, M., Chai, J.Y.: Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems* 20(6), 511–526 (2007)
5. Chai, J.Y., Jin, R.: Discourse structure for context question answering. In: Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004, pp. 23–30 (2004)
6. Kirschner, M., Bernardi, R., Baroni, M., Dinh, L.T.: Analyzing interactive QA dialogues using logistic regression models. In: Serra, R., Cucchiara, R. (eds.) *AI*IA 2009. LNCS (LNAI)*, vol. 5883, pp. 334–344. Springer, Heidelberg (2009)
7. Rieser, V., Lemon, O.: Learning effective multimodal dialogue strategies from Wizard-of-Oz data: Bootstrapping and evaluation. In: Proceedings of ACL, pp. 638–646 (2008)
8. Van Schooten, B.W., Op den Akker, R., Rosset, S., et al.: Follow-up question handling in the IMIX and Ritel systems: A comparative study. *Natural Language Engineering* 15(01), 97–118 (2009)
9. Kelly, D., Lin, J.: Overview of the TREC 2006 ciQA task. *ACM SIGIR Forum* 41(1), 107–116 (2007)
10. Dang, H.T., Lin, J., Kelly, D.: Overview of the TREC 2007 question answering track. In: Proceedings of TREC, vol. 2007(5.3), p. 3 (2007)
11. Hickl, A., Harabagiu, S.: Enhanced interactive question-answering with conditional random fields. In: Proc. of the IQA Workshop at HLT-NAACL, pp. 25–32. Association for Computational Linguistics, New York City (2006)
12. Misu, T., Georgila, K., Leuski, A., et al.: 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, pp. 84–93. Association for Computational Linguistics (2012)
13. Rzeniewicz, J., Szymański, J., Duch, W.: Adaptive Algorithm for Interactive Question-Based Search. In: Shi, Z., Leake, D., Vadera, S. (eds.) *IIP 2012. IFIP AICT*, vol. 385, pp. 186–195. Springer, Heidelberg (2012)
14. Burke, R.D., Hammond, K.J., Kulyukin, V., et al.: Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine* 18(2), 57 (1997)
15. Pal, S., Bhattacharya, S., Datta, I., et al.: A Framework for Automatic Generation of Answers to Conceptual Questions in Frequently Asked Question (FAQ) Based Question Answering System. *International Journal of Advanced Research in Artificial Intelligence* 1(2) (2012)
16. Cong, G., Wang, L., Lin, C.Y., et al.: Finding question-answer pairs from online forums. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 467–474. ACM (2008)

17. Cao, Z., Qin, T., Liu, T.Y., et al.: Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th International Conference on Machine Learning, pp. 129–136. ACM (2007)
18. Liu, T.Y.: Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3(3), 225–331 (2009)
19. Verberne, S.: In search of the why: Developing a system for answering why-questions. *Sn, SI* (2010)
20. 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)
21. Yue, Y., Finley, T., Radlinski, F., et al.: A support vector method for optimizing average precision. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 271–278. ACM (2007)
22. Furey, T.S., Cristianini, N., Duffy, N., et al.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16(10), 906–914 (2000)
23. Mood, A., Graybill, F., Boes, D.: *Introduction to the Theory of Statistics*, 3rd edn. McGraw-Hill (1974)
24. Recall, Z.M.: Precision and average precision. Department of Statistics and Actuarial Science. University of Waterloo, Waterloo (2004)
25. Dong, Z., Dong, Q.: *HowNet* (2000)
26. Liu, Q., Li, S.J.: Word semantic similarity computation based on HowNet. In: Proc. 3rd Chinese Word Semantic Conference (2002)
27. Blunsom, P., Kocik, K., Curran, J.R.: Question classification with log-linear models. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 615–616. ACM (2006)
28. 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)

Document Oriented Gap Filling of Definite Null Instantiation in FrameNet

Ning Wang¹, Ru Li^{1,2}, Zhangzhang Lei¹, Zhiqiang Wang¹, and Jingpan Jin³

¹School of Computer & Information Technology

²Key Laboratory of Ministry of Education for Computation Intelligence & Chinese Information Processing, Shanxi University, Taiyuan 030006, China

³School of Computer, Beijing Institute of Technology, Beijing 100081, China

{Wanganning.com, Leizhangzhang77777, zhiq.wang, jinjingpan}@163.com, liru@sxu.edu.cn

Abstract. Null instantiation has attracted much attention recently. In this paper, we focus on gap filling of definite null instantiation, namely, finding an antecedent for a given definite null instantiation from context. Most of the approaches for solving this problem use syntactic features, and only few consider semantic features. Moreover, these approaches only take the noun, noun phrase and pronoun as candidate words, so the coverage of antecedent is narrow. In this paper, we use new features of words and frame except traditional features, and create a rule to build candidate words set. At last, we choose the best candidate words set and feature template based on employing standard annotated corpus, then use them to deal with corpus of NIs only in task SemEval-10 Task 10. According to the experimental results, our approach achieves a better performance than existing approaches.

Keywords: Definite Null Instantiation, Gap Filling, Semantic Features, Candidate Words Set.

1 Introduction

FrameNet [1] is a computational lexicography project, which based on the theory of Frame Semantics and concerned with networks of meaning in which words participate. The primary units of lexical analysis in FrameNet are the frame and the lexical unit. Null instantiation is the core frame element which is neither expressed as a dependent of the predicator nor can it be found through gap filling in FrameNet [2]. We can divide null instantiation into two categories: definite null instantiation (DNI) and indefinite null instantiation (INI). Cases of indefinite null instantiation are the missing objects of verbs like eat, sew, bake, and drink, etc. where the nature or semantic type of the missing element can be understood, and there is no need to retrieve or construct a specific discourse referent, as core frame element FOOD in the following 1. Definite null instantiation are those in which the missing element must be something that is already understood in the linguistic or discourse context, as the following example in 2. The target word difficult evokes the difficulty frame, which has two core frame

elements, only one of which is filled locally, namely ACTIVITY, which is realized by business. However, another argument, EXPERIENCER, is filled by the I in preceding sentence.

1. [Sue INGESTOR]had eaten already.[INI FOOD]
2. I think that I shall be in a position to make the situation rather more clearly to you before long. It has been an [exceedingly DEGREE] difficult and most complicated [business ACTIVITY].[DNI EXPERIENCER]

Gap filling identifies the overt antecedents of null instantiation in controlled structures, as INIs do not need to be accessible within a context, the task of resolving NIs is restricted to DNIs. As the example 2, gap filling of DNI aims to find the overt expression “I” to fill the omitted frame element “EXPERIENCER”. Because DNI is not overt argument in sentence, it is difficult to find some information to describe it, which causes the gap filling of DNI becomes a challenging problem in discourse processing.

Given a DNI, we think that gap filling of DNI can be seen as a classified problem to judge whether a candidate could be taken as filler of a DNI, so we use classification method to solve the problem. In this task, an important step is to determine the scope of candidate words set and features for classification. In this paper, we design a rule to select candidate words set, combine features in a diversified portfolio, and finally use the maximum entropy model to classify candidate words.

The remainder of this paper is structured as follows. In section 2, we briefly summarize the related work on gap filling of DNI. Section 3 introduces the way to build select rule of candidate words set, features description and the maximum entropy model in DNI gap filling. Section 4 reports the results of experiments. Finally, Section 5 concludes this paper.

2 Related Work

There is a growing interest in developing algorithms for resolving null instantiations. Null instantiations were the focus of the SemEval-10 Task 10, which showed two mission modes, namely full task (semantic role recognition and labeling + NI linking) and NIs only task, i.e. the identification of null instantiations and their referents given a test set with gold standard local semantic argument structure[3]. This paper focus on NIs only task to realize gap filling of DNI.

There are two teams participate in NIs only task. Tonelli and Delmonte[4] developed a knowledge-based system called VENSES++, different resolution strategies are employed for verbal and nominal predicates. For verbal predicates, the system finds a comparable PAS in previous sentences, and then looks for the best head available in that PAS as a referent for the DNI in the current sentence by semantic matching with the FE label. For nominal predicates, NIs are resolved by making use of a common sense reasoning module that builds on ConceptNet[5]. Because it relies on large-scale corpus to train the feature templates, ultimately they obtained precision and recall rate was 4.62% and 0.86%. The second SemEval system[6] modeled the problem as the same way of semantic role labeling. They consider nouns, pronouns, and noun phrases from the previous three sentences as candidate DNI referents. When evaluating

potential filler, the system checks whether it fills the null instantiated role overtly in one of the FrameNet sentences at first, if not, they calculate the distributional similarity between filler and role. But, these semantic features have virtually no effect on performance possibly due to data sparseness.

Philip and Josef[7] developed a weakly supervised approach that investigates and combines a number of linguistically motivated strategies. Silberer and Frank[8] view NI resolution as a coreference resolution (CR) task, employing an entity-mention model, combining features of SRL and CR, and achieving F-score is 7.1% at last. Gerber and Chai[9,10] present a study of implicit arguments for a group of nominal predicates. They also use an entity mention approach and model the problem as a classical supervised task, implementing a number of syntactic, semantic, and discourse features. Because Gerber and Chai’s corpus cover 10 nominal predicates from the commerce domain, with on average 120 annotated instances per predicate, so their results are noticeably higher than those obtained for the SemEval data.

3 Model for Gap Filling of DNI

It is critical to determine search space and POS of candidate fillers for DNI in gap filling of DNI, as well as features for classification. Search space is the number of sentences that candidate fillers away from target, the choice of search space could affect the cover probability of antecedent and the result of DNI gap filling. A good candidate words set (includes search space and POS of words) could reduce the complexity of the system and improve the efficiency of the experiment. In this section, we focus on the selection of candidate words set and features.

3.1 Selection of Candidate Words Set

Candidate words are those which may be used as explicit referents of implicit argument. The accuracy of search space and POS for candidate words would influence the result of gap filling. Because the distribution of explicit referents for DNI is chaotic, and their part-of-speech is diverse, it is difficult to create an appropriate candidate words set which could maximum cover the entire antecedent and has a minimum size. In order to solve this problem, we count the distribution of DNI referents in training data of NIs only task.

Table 1. The distribution of DNI referents in training data

Distance of sentences	0	1	2	3	4	5	6	7	8
number	95	63	19	6	4	5	4	2	2

Table 1 shows the main distribution of DNI referents in training data. We can see that the distribution of DNI referents mainly concentrates in the same sentence, previous one sentence and two sentences, and other sentences are relatively less. The data listed in table 1 account for only 65.79 percent of the total number of DNI referents. In training data, there are 58 DNIs have no referent, 6 appear in six sentences before, 28 appear at least 25 sentence prior. Observe the above data, we can draw that

the coverage probability has obvious growth trend from one sentence to three sentences, when we choose four sentences or five sentences as search space, there are only 1 percent increase than others. Based on the above data, we list several methods in table 2 to choose the best candidate words set.

Table 2. The search space of candidate words set (%)

Num	Search space	coverage probability	description
H1	$n \leq 2 \ \&\&n \neq 0$	46.05	Words in previous two sentences
H2	$n \leq 2 \ \&\&n = 0$	77.30	Words in this and previous two sentences
H3	$n \leq 3 \ \&\&n \neq 0$	48.03	Words in previous three sentences
H4	$n \leq 3 \ \&\&n = 0$	79.28	Words in this and previous three sentences
H5	$n \leq 4 \ \&\&n \neq 0$	49.34	Words in previous four sentences
H6	$n \leq 4 \ \&\&n = 0$	80.59	Words in this and previous four sentences
H7	$n \leq 5 \ \&\&n \neq 0$	50.99	Words in previous five sentences
H8	$n \leq 5 \ \&\&n = 0$	82.24	Words in this and previous five sentences

In corpus, some words in search space are impossible to act as frame element fillers, such as VB, VBP and VBZ. These words would increase the complexity of the system and impact the efficiency of the experiment, therefore we should remove them from candidate words set. In the following, we analyze the part-of-speech distribution of DNI referents in training data to choose suitable candidate words.

Table 3. Part-of-speech distribution of DNI referents in training data (%)

POS of antecedent	NPB	PRP	NNP	NP	S	PRP\$	VP	NN	VBN	SBA	SG
Num	91	77	15	13	12	9	7	3	1	1	1
Probability	39.57	33.48	6.52	5.65	5.22	3.91	3.04	1.30	0.43	0.43	0.43

As shown in table 3, the words which POS are NPB (noun phrase) and PRP (pronoun) account for 73.05% in total, as a result we take them as basic POS of candidate words for DNI, and the following rules are devised for building candidate words set based on the data in table 3:

1. Given the current DNI frame element, looking for the same frame elements in the train data.
2. If the same frame elements are found, counting the POS of their fillers, choosing the largest one as C, and taking NPB, PRP, and C as candidate words for DNI in search space.
3. Otherwise, only taking noun phrases and pronoun as candidate words for DNI in search space.

3.2 Features Description

Feature selection is important in classification problems and the performance of classification largely depends on feature selection, which is also a key issue in gap filling of DNI. For definite null instantiations, their conceptually-relevant content is left unexpressed or is not explicitly linked to the frame via linguistic conventions, so it is difficult to get some information from discourse to describe them. Only can we take as features for gap filling of DNI are information of candidate words and DNI frame element.

In discourse, head words are frequently used as role filler. The closer head word away from the target, the more likely it becomes DNI explicit expression. Hence, we take information of head word as features for gap filling of DNI. In a frame, the NI type of the same frame element would be different for the lexical varies, and different roles would be having different NI type under the same lexical as well. In the case of frame element GOAL and SOURCE, some verbs allow its omission under indefinite null instantiation (1, 4), others allow its omission under definite null instantiation (2, 3).

1. Adam **left** Paris [INI Goal].
2. Smithers **arrived** [DNI Goal].
3. Sue **left** [DNI Source].
4. Sue **arrived** in Rome [INI Source].

Table 4. Features description

	Num	Features Name	Features description
C1	T1	DistantSen	The number of sentences between candidate and target
	T2	WordContent	Candidate word
	T3	WordCat	Cat of candidate word
	T4	WordLength	Length of candidate word
C2	T5	headWord	Head word of candidate word
	T6	headWordLemma	The lemma of head word
	T7	HeadWordPos	The pos of head word
C3	T8	frame	The frame that target evokes
	T9	FENI	DNI argument
	T10	target	target
	T11	targetLemma	The lemma of target
	T12	targetPos	The pos of target

In conclusion, we also take account of frame information when gap filling of DNI. In table 4, we describe all of the features that may be useful in gap filling for DNI.

3.3 Maximum Entropy Models

Maximum entropy model which is based on the maximum entropy principles is set up for all known facts without any other influence of factors. We can add any useful feature for the final classification without considering the interaction between each other. Maximum entropy model, as a kind of statistical method, has been widely used to aspects of natural language processing (such as part-of-speech, chinese word segmentation and machine translation) in the late.

In the experiment, it will involve a variety of factors when predicting whether a candidate is DNI filler. Supposed X is a vector of these factors, y represents whether potential filler is DNI referent or not. $p(y | X)$ is a probability that a candidate is predicted as filler of DNI. Maximum entropy model ask for $p(y | X)$ to make the entropy defined below largest under certain restrict conditions.

$$H(p) = - \sum_{X,y} p(y | X) \log p(y | X)$$

The restrict conditions refers to all known facts actually, the final output of the probability is:

$$p(y | X) = \frac{1}{Z(X)} \exp \left(\sum_i^n \lambda_i f_i(X, y) \right),$$

$$Z(X) = \sum_y \exp \left(\sum_i^n \lambda_i f_i(X, y) \right)$$

$f_i(X, y)$ is features of maximum entropy model, n is the number of features, and the features describe the relationship between X and y . λ_i is the weight of each feature.

In this paper, we use the maximum entropy toolkit of Dr. Zhang Le for classification[11].

4 System Output and Evaluation

In gap filling of DNI, search space of candidate words set and feature selection are two key steps in the experiment. In this section, we use corpus which has annotated NI type to get the best feature template firstly, and then choose the best candidate words set with the best feature template in the same corpus. At last we apply them to NIs only task data and compare the result with previous works.

4.1 Corpus

In our experiment, we used the corpus distributed for SemEval-2010 Task 10 on “Linking Events and Their Participants in Discourse”. The data set consists of the SemEval-2007 data plus annotated data in the fiction domain: parts of two Sherlock Holmes stories by Arthur Conan Doyle. The training set has about 7800 words in 438 sentences; it has 317 frame types, including 1370 annotated frame instances. The test set consists of two chapters, which has about 9000 words, 525 sentences, 452 frame types and 1703 frames. All data released for the 2010 task include part-of-speech tags, lemmas, and phrase-structure trees from a parser, with head annotations for constituents. Table 5 shows the statistics about this data set.

Table 5. Statistics for NIs only task corpus

Data Set	sentences	frame inst.	frame types	DNIs
train	438	1370	317	304
test	525	1703	452	349

4.2 Evaluation Measures

The correct gap filling of DNI refers to the content and boundary of antecedent correct, as well as NI type, we use precision, recall, and F-score to evaluate the performance of this system. Assume that C_p is the DNI number predicted by system, C_c is the DNI number which is predicted correct and DNI number in the answer of test set for C_o , and then we define precision, recall and F-score as following formulas.

$$P = \frac{C_c}{C_p} \quad R = \frac{C_c}{C_o} \quad F = \frac{2PR}{P + R}$$

We evaluate the performance of experiments based on their average value of chapter 13 and chapter 14.

4.3 Result in Gold Standard Annotated Corpus

In this section, we use the corpus which has annotated information about null instantiation, i.e., the NI type (DNI vs. INI), assuming that NIs have been identified and correctly classified as DNI or INI, we only focus on the DNI. For each DNI, the experiment chooses candidate words in context based on the rules defined in 3.1, and then takes their features as input for training and predicting on the maximum entropy model. We think a DNI has no referents, when no word in candidate words set of this DNI is judged as its antecedent. In order to get the best feature template, we choose H3 in table 1 as search space for DNI candidate set according to Chen et al. The results are listed in table 6.

Table 6. The results of different characteristics combination under gold standard annotation(%)

character	Chapter13			Chapter 14		
	Prec.	Rec.	F	Prec.	Rec.	F
C1	22.93	22.78	22.86	28.04	27.75	27.89
C2	22.93	22.78	22.86	28.04	27.75	27.89
C3	16.98	22.78	19.46	24.32	28.27	26.15
C1+C2	22.93	22.78	22.86	28.04	27.75	27.89
C1+C3	23.27	23.42	23.34	27.55	28.27	27.91
C2+C3	18.09	22.78	20.17	24.76	27.23	25.94
C1+C2+C3	23.27	23.42	23.34	27.69	28.27	27.98

Based on the data shown in table 6, we can get that combination of C1, C2 and C3 has better performance than others. This means that combining all features to build feature template could provide more information to the system. In addition, table 6 also shows that the results of chapter 13 were lower than chapter 14, which may be caused by several reasons. Firstly, in test data, chapter 13 contains 97 DNI frame elements which are same with train data, while chapter 14 has 130. So it is obvious that candidate words set in chapter 14 can cover more DNI referents than chapter 13 based on the first candidate words select rule. Secondly, the experiment consider words in previous three sentences as candidate DNI referents, but there are 14 percent of antecedent out of it in chapter 13, and 5 percent in chapter 14. A case in chapter 13 is given as follows:

```
<fe id="s42_f2b_e1" name="Judge">
  <fenode idref="s33_8" />
  <flag name="Definite_Interpretation" />
</fe>
```

Finally, it exists that DNI referents is composed of multiple phrases rather than one word, which is not taken into account in the system. This situation in chapter 13 has 8, and in chapter 14 has 7. For example:

```
<fe id="s33_f6_e2" name="Action">
  <fenode idref="s33_7" />
  <fenode idref="s33_13" />
  <fenode idref="s33_12" />
  <fenode idref="s33_11" />
  <fenode idref="s33_9" />
  <fenode idref="s33_8" />
  <flag name="Definite_Interpretation" />
</fe>
```

Because combining all features to build feature template could improve the performance of the system, so we choose C1+C2+C3 in table 6 as features to study the influence of candidate words set in different search space, aiming to choose the best one which could get optimal performance. The results are showed in table 7.

When choose H2、H4、H6 or H8 as search space of candidate words set, we can get more information to train than others. It leads to the number of classification results and correctly predicted more than choose H1、H3、H5 or H7. As a result, the precision of the former ones is lower than the latter ones, but have higher recall. We can conclude that the F-score of system is highest when candidate set is H3 via comparison.

Table 7. The results of DNI gap filling in different candidate words sets (%)

Num	Prec.	Rec.	F
H1	25.24	25.53	25.38
H2	24.21	27.06	25.55
H3	25.48	25.84	25.66
H4	23.40	26.74	24.95
H5	25.31	25.53	25.42
H6	23.23	26.42	24.72
H7	25.31	25.53	25.42
H8	22.72	26.79	24.57

4.4 Result in NIs only Task Test Data

We have systematically evaluated our model on the corpus distributed for NIs only task of SemEval-10's Task-10, as described in Section 4.1. Besides, in order to focus on gap filling of DNI automatically and compare with related work, all the experiments are carried out on gold-standard semantic role labeling. The complete task can be modeled as a pipeline consisting of three sub-tasks: (a) identifying potential NIs by taking into account information about core arguments, (b) automatically distinguishing between DNIs and INIs via maximum entropy model, and (c) resolving NIs classified as DNI to a suitable referent in the text. We identify NI types based on FrameNet and use maximum entropy model to classify Nis[12]. The result of DNI identification is shown in table 8. The number of our predicted DNI is more than VENSES++, which is a big reason why our result of DNI gap filling is better than them. But we can also see from the table, our result is far from the gold standard number. Because task (c) is on the basis of task (a) and (b), so it is a limit to the result of DNI gap filling.

Table 8. Result of NI identification

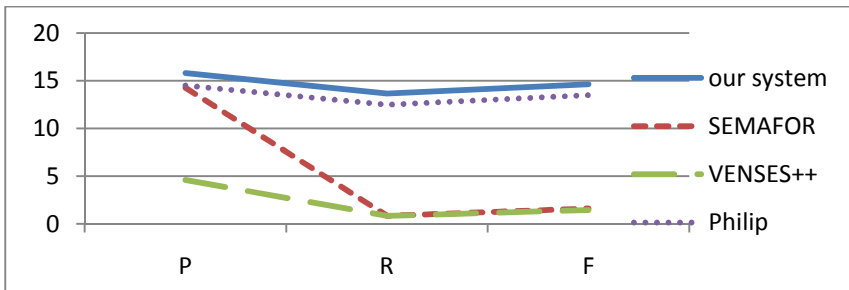
	Chapter 13		Chapter 14	
	DNIs	INIs	DNIs	INIs
Gold	158	116	191	245
VENSES++	35	16	30	20
Predicted	144	85	158	144

As concluded in section 4.3, the system achieved the best performance when the model was H3+C1+C2+C3, so we use it to build feature template for gap filling of DNIs which are predicted by our system.

Table 9. Compare the results in corpus of gold standard annotated and NIs only task (%)

corpus	gold standard annotated corpus			NIs only task corpus		
	Prec.	Rec.	F	Prec.	Rec.	F
Chapter 13	23.27	23.42	23.34	13.89	12.67	13.25
Chapter 14	27.69	28.27	27.98	17.72	14.66	16.05
Average	25.48	25.85	25.66	15.81	13.66	14.65

Table 9 shows the average result of gap filling of DNI in gold-standard annotated corpus and NIs only corpus. We can see from the table that precision of the former is nearly 10 percent higher than the latter, as well as recall and F-score, the majority reason is that the result of third step in NIs only corpus is greatly influenced by the former two steps. According to our statistics, the number of DNI predicted by the system accounts for 66.76 percent of the answer, and the number that predicted correctly is only 42.41 percent, which could cause that the input of DNI gap filling in NIs only task is little than it in gold-standard annotated corpus, which would largely influence on the result of the classification.

**Fig. 1.** Comparison with previous works

We compare our results with precious work to illustrate the effectiveness of our model. The comparison is showed in figure 1, the horizontal axis display precision, recall and F-score of every system, and the ordinate said percentage. We can see from the figure that our system is better than other ones, the reason of which may boil down to the following:

1. SEMAFOR and VENSES++ combine classification of NI and DNI resolution, they look for an antecedent for an omitted role, if find it, they label the role as DNI, otherwise, it is labeled as INI. While in our system, we decompose the problem into two independent steps. Our system identifies null instantiation at first, and then resolves the DNIs, which entails finding referents in the context. By the way, we can take the DNIs which have no referent into account, so the recall of our system is higher than others.
2. SEMAFOR system consider nouns, pronouns, and noun phrases from the previous three sentences as candidate DNI referents, so 26.65 percent of gold DNI referents haven't be considered according to table 1 and table 2. In addition, the semantic features they choose received negligible weight and had virtually no effect on performance because of data sparseness.
3. VENSES++ system requires large corpus to get information of PAS and AHDS, but the corpus of NIs only task is too small to cover all the information.
4. Philip and others only make use of minimal supervision for modeling the role linking task, which make their result lower than ours.

Compared with the three models, there are two advantages of our proposed model. One is the rule for selecting candidate words in this paper could maximum cover all the DNI referents. And the other one is adding information of head word and frame to traditional features could get the best feature template.

5 Conclusion and Further Work

In this paper, we have presented a new approach to find the antecedents for definite null instantiations which are widely used in many fields of natural language processing. By adding new features such as the information of head word and frame to traditional features, we proposed a candidate selection rule which can be used to choose the best candidate words set and combination of features. Experiments show that the proposed model can get a better result than existing ones. It is our wish that this study provides new views and thoughts in natural language processing.

Identification and classification of null instantiation is the cornerstone of DNI gap filling, so it is significant to improve the performance of NI classification for DNI gap filling. Besides, there are a lot of relations between frames in FrameNet. If the relationship of two frames is inheritance, their frame element fillers also have some special connection. Therefore, we will focus on the research of applying frame relations to gap filling of DNI in the future.

Acknowledgments. This work was supported by National Natural Science Fund of China (No. 60970053), National Language Committee "1025" planning research (No. YB125-19), International cooperation in science and technology project of Shanxi Province (No. 2010081044), National 863 plans projects (No. 2006AA01Z142) and Research Project Supported by Shanxi Scholarship Council of China(No. 2013-015).

References

1. Fillmore, C.J., Johnson, R., Petruck, M.R.L.: Background to FrameNet. *International Journal of Lexicography* 16(3), 235 (2003)
2. Fillmore, C.J., Ruppenhofer, J., Baker, C.F.: Framenet and representing the link between semantic and syntactic relations. In: Huang, C., Lenders, W. (eds.) *Frontiers in Linguistics. Language and Linguistics Monograph Series B*, vol. I, pp. 19–59. Institute of Linguistics, Academia Sinica, Taipei (2004)
3. Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., Palmer, M.: SemEval-2010 Task 10: Linking Events and Their Participants in Discourse. In: *Proc. of the HLT-NAACL Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, Boulder, Colorado (2009)
4. Tonelli, S., Delmonte, R.: VENSES++: Adapting a deep semantic processing system to the identification of null instantiations. In: *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, Uppsala, Sweden*, pp. 296–299 (2010)
5. Liu, H., Singh, P.: ConceptNet: a practical commonsense reasoning toolkit (2004), <http://web.media.mit.edu/~push/ConceptNet.pdf>
6. Chen, D., Schneider, N., Das, D., Smith, N.A.: SEMATOR: Frame Argument Resolution with Log-Linear Models. In: *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, Uppsala, Sweden*, pp. 264–267 (2010)
7. Gorinski, P., Ruppenhofer, J., Sporleder, C.: Towards weakly supervised resolution of null instantiations. In: *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pp. 119–130 (2013)
8. Silberer, C., Frank, A.: Casting implicit role linking as an anaphora resolution task. In: *Proceedings of the First Joint Conference on Lexical and Computational Semantics: Proceedings of the Main Conference and the Shared Task: Proceedings of the Sixth International Workshop on Semantic Evaluation*, vol. 1, 2, pp. 1–10. Association for Computational Linguistics (2012)
9. Gerber, M., Chai, J.Y.: Beyond nombank: a study of implicit arguments for nominal predicates. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010*, pp. 1583–1592. Association for Computational Linguistics, Stroudsburg (2010)
10. Gerber, M., Chai, J.Y.: Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics* 38(4), 755–798 (2012)
11. Le, Z.: Maximum entropy modeling toolkit for python and c++ (2005), http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html
12. Lei, Z., Wang, N., Li, R., Wang, Z.: Definite Null Instantiation Recognizing in FrameNet. *Journal of Chinese Information* 27(3), 107–112 (2013)

Interesting Linguistic Features in Coreference Annotation of an Inflectional Language*

Maciej Ogrodniczuk¹, Katarzyna Głowińska², Mateusz Kopeć¹,
Agata Savary³, and Magdalena Zawisławska⁴

¹ Institute of Computer Science, Polish Academy of Sciences

² Lingventa

³ François Rabelais University Tours, Laboratoire d'informatique

⁴ Institute of Polish Language, Warsaw University

Abstract. This paper reports on linguistic features and decisions that we find vital in the process of annotation and resolution of coreference for highly inflectional languages. The presented results have been collected during preparation of a corpus of general direct nominal coreference of Polish. Starting from the notion of a mention, its borders and potential vs. actual referentiality, we discuss the problem of complete and near-identity, zero subjects and dominant expressions. We also present interesting linguistic cases influencing the coreference resolution such as the difference between semantic and syntactic heads or the phenomenon of coreference chains made of indefinite pronouns.

1 Introduction

For languages still lacking state-of-the-art coreference resolution tools, manual annotation of coreference over a substantially large dataset is traditionally the first step of the work: after the labor-intensive process is over, a supervised resolver can be trained on the hand-annotated documents. Since such resource was until recently unavailable for Polish, all coreference-related work concentrated on theoretical modelling, rule-based or projection-based approaches, and were evaluated on very small data samples.

All the issues above were highly motivating for creation of the first large-scale corpus of general direct nominal coreference of Polish (currently in last phases of construction). In this paper we present the decisions we made while selecting and adopting the annotation schema for this corpus and how they were influenced (and then verified against the real-world data) by recent works on the subject and our understanding of certain linguistic phenomena related to anaphora and coreference in highly inflectional languages.

Based on empirical data collected in the process of the corpus creation, we discuss how certain linguistic features of an inflectional language influence the annotation

* The work reported here was carried out within the *Computer-based methods for coreference resolution in Polish texts (CORE)* project financed by the Polish National Science Centre (contract number 6505/B/T02/2011/40). The paper is also co-funded by the European Union from resources of the European Social Fund, Project PO KL "Information technologies: Research and their interdisciplinary applications".

schema and resolution tools. Statistics of the respective phenomena and inter-annotator agreement values are also presented.

2 Mentions and Coreference Clusters

Our annotation schema defines mentions as nominal groups (NGs) taking into account their *potential* referentiality, which is based on observation that certain stylistically marked cases allow of using traditionally non-referential expressions in referential contexts, as in (1)¹.

- (1) *Nie wahał się włożyć kij w mrowisko.*
Mrowisko to, czyli cały senat uniwersytecki, pozostawało zwykle niewzruszone.
'He didn't hesitate to put a stick into an anthill (i.e. to provoke a disturbance).
This anthill, i.e. the whole university senate, usually didn't care.'

The following phrase types are treated as NGs:

1. nouns, nominal phrases, personal pronouns,
2. numeral groups (*trzy rowery* = 'three bicycles'),
3. adjectival phrases with elided nouns (*bukiet z czerwonych kwiatów i z tych niebieskich* = 'a bouquet of the red flowers and these blue ones'),
4. date/time expressions of various syntactic structures,
5. coordinated nominal phrases, including conjoining commas (*krzesło, stół i fotel* = 'a chair, a table, and an armchair').

The boundaries of mentions are set to involve as broad contexts as possible to maximally disambiguate entities (to refer to 'the car which hit my wife', not just 'the car'). Elements allowed within mention contents are:

1. adjectives and adjectival participles in agreement (with respect to case, gender and number) with superior noun (*duży czerwony tramwaj* = 'big red tram'),
2. subordinate nouns in the genitive case (*kolega brata* = 'my brother's colleague'),
3. nouns in apposition (*malarz pejzażysta* = 'landscape painter', pol. 'painter landscapist'),
4. subordinate prepositional-nominal phrases (*koncert na skrzypce i fortepian* = 'a concerto for violin and piano'),
5. relative clauses (*dziewczyna, o której rozmawiamy* = 'the girl that we talk about').

The deep structure of NGs, i.e. all embedded phrases not containing finite verb forms having semantic heads other than those of the superior phrase (which reference different entities), are annotated, therefore the fragment *dyrektor departamentu firmy* 'manager of a company department' contains 3 nominal phrases, referencing the manager of a company department ('*dyrektor departamentu firmy*'), the company department ('*departamentu firmy*') and the company ('*firmy*') alone.

This assumption is also valid for coordination — we annotate both the individual constituents and the resulting compound, because they can be both referred to:

¹ Henceforth, we will mark coreferent NGs with (possibly multiple) underlining, and non-coreferent NGs with dashed underlining.

- (2) *Jan z Marią przyszli na obiad. Oni są przemili, zwłaszcza Maria.*
 ‘Jan and Maria have come to dinner. They are charming, especially Maria.’

Discontinuous phrases and compounds are also marked:

- (3) *To był delikatny, że tak powiem, temat.*
 ‘It was a touchy, so to speak, subject.’

Zero anaphora, very frequent in Polish due to rich inflection of verbs, is marked by including verbs (whose pronominal subjects are elided) into coreference clusters, as in (4). Zero anaphora is not considered for objects and complements.

- (4) *Maria wróciła już z Francji. ØSpędziła tam miesiąc.*
 ‘Maria came back from France. ØHad_{singular:feminine} spent a month there.’

Coreference clusters group mentions referring to the same discourse-world entity. In our task we concentrate on identity of reference in its strict form (direct reference) with an extension of so called near-identity (see section 4).

3 Related Work

The annotation schema resumed in the previous section was presented in details in [20]. It was also compared with several approaches to coreference annotation in languages that show coreference-relevant morphosyntactic similarities with Polish, i.e. Slavic languages [1,2] and Spanish [3,4] due to its frequent zero subject. A recent study dedicated to English [5] was also considered for obvious dominance reasons in NLP. In view of this contrastive study our annotation schema shows three major novel aspects which we deeply analyse in this paper:

- large-scale experiments with near-identity,
- introduction of dominant expressions,
- pointing at semantic rather than syntactic heads.

To a lesser extent, the fact of systematically taking zero subjects into account in our approach brings some new insights into the state of the art. In order to further verify these novelty issues, we present below some other bibliographic references reporting on coreference annotation schemas which were applied to corpora of about 200 thousand words or more — according to [6], p. 10.

The series of ACE (Automatic Content Extraction) program has been carried out from 1999 to 2008 for a varying number of languages, including Arabic, Chinese, English, and Spanish. It was meant to boost the development of automatic detection and characterization of meaning conveyed by human texts. The ACE-2007 annotation guidelines for Spanish [7] gives the rules of annotating and disambiguating entities. The entity typology is rather fine-grained: it consists of 7 main types (person, organization, geopolitical entity, etc.) and several dozens of subtypes (individual, group, governmental, commercial, etc.). Two coreference relations are considered: identity and apposition. The former is further subdivided into generic and non-generic. Mentions are

NGs (including attached prepositional phrases and relative clauses) and can be nested (*the president of Ford*). Each NG should have its (syntactic) head marked. Heads are marked but their definition is confusing (the syntactic head can be multi-word, and then its last token is marked). Semantic heads different from syntactic ones are not an issue. The problem of Spanish zero subject is not mentioned.

[8] describe the annotation of the 22-thousand-sentence Tübingen treebank of German newspaper texts (TüBa-D/Z) with a set of 7 coreference relations (coreferential, anaphoric, cataphoric, bound, split antecedent, instance, and expletive). These are no equivalence relations: they are non-symmetric and mostly non-transitive, thus they do not divide the set of referents into disjoint clusters. For instance, the split antecedent relation holds between a plural expression and a mention of a single member. E.g. in '*John and Mary were there... Both...*', *John* and *both*, as well as *Mary* and *both* are coreferential but *John* and *Mary* are not. Potential markables are definite NGs, personal pronouns, relative, reflexive, and reciprocal pronouns, demonstrative, indefinite and possessive pronouns. All of them correspond to nodes of the already existing parse trees resulting from prior syntactic annotation. Unlike in our approach, predicative nominal groups (NGs) seem to be considered coreferential with subjects. Zero subjects are not an issue in German. Neither dominant expressions nor semantic heads are mentioned.

[9] address the construction of a 182-thousand-word Italian Content Annotation Bank of newspaper texts (I-CAB). Mentions are NGs, possibly containing modifiers, prepositional complements or subordinate clause, representing entities (persons, organizations, etc.) or temporal expressions. ACE-2003 annotation guidelines for English are adopted and extended, to cover notably clitics contiguous with verbs (*vederlo*) and coordinated expressions (*John and Mary*). The paper announces future annotation of relations between entities but it is unclear where its results have been described.

[10] present NAIST, a 38-thousand sentence Japanese corpus annotated for coreference and predicate-argument relations (including nominal predicates relating to events). They consider identity-of-reference relations for the former, and both identity-of-reference and identity-of-sense relations for the latter. They pay a special attention to zero anaphora, whose role — not only as a subject but as an object or a complement as well — is particularly visible when coreference and predicates' arguments are annotated jointly. Namely, the frames for elided arguments have to be filled out with antecedents appearing in other sentences than the predicate itself. The reported inter-annotator agreement for coreference annotation is 0.893 for recall and 0.831 for precision. No mention of dominant expressions or semantic heads is made.

[11] describe OntoNotes, a system of multi-layered annotated corpora in English, Chinese and Arabic. It is supposed to make up for the drawbacks of previous annotation schemata, mainly MUC and ACE in that the coreference annotation is not restricted to NGs and a larger set of entity types is considered. The English corpus consists of 300-thousand-word newspaper texts, later completed by broadcast conversation data [12]. All data have been previously annotated for syntax, verbal arguments and word senses (by linking words to nodes of an external ontology). Thus, mention candidates correspond to nodes of pre-existing syntax trees. As in ACE, two coreference relations are considered: identity and apposition. The main mention candidates are specific

NGs, pronouns (*they*, *their*) and single-word verbs coreferent with noun phrases (e.g. *the sales rose by 18% ← the strong growth*). Expletive (*it rains*), pleonastic (*there are*) and generic (*you need*) pronouns are not marked. Generic, underspecified or abstract entities are only partly considered by identity: those cannot be interlinked among themselves, even if they can be linked with referring pronouns (*parents ← they*). Nested structures are generally marked but exceptions occur in dates (e.g. no subphrase of *Nov. 2, 1999* is coreferent with *November*). Only intra-document coreference is annotated, thus dominant expressions (motivated in our approach notably by future inter-document coreference annotation) are not an issue. Zero subjects are addressed with respect to pro-drop Arabic and Chinese pronouns [12]. Since such pronouns are materialized in parse trees as separate nodes, their inclusion in coreference chains is straightforward. The existence of semantic heads different from syntactic ones is not mentioned.

[13] describe a 200-thousand-word coreference-annotated corpus of Dutch newspaper texts, transcribed speech and medical encyclopedia entries. Its annotation schema is largely based on the MUC-7 annotation guidelines for English². Annotation focuses mainly on identity relations between NGs but other non-equivalence relations are also introduced: bound relations (*everybody ← they*), bridging relations (e.g. superset-subset or group-member), and predicative relations (e.g. *John* is a *painter*). Syntactic heads are pointed at but semantic heads different from syntactic ones do not seem to be an issue (e.g. *tons* is the head of *200,000 tons of sugar*). The ideas of dominant expressions and zero subjects are not present. Predicative NGs and appositions are considered as mentions coreferent with their subjects. Discontinuous NGs are taken into account. The inter-annotator agreement measured as the MUC-like F-score, is 0.76 for identity relations, 0.33 for bridging relations and 0.56 for predicative relations.

The above bibliographic study confirms the novelty of annotating at least three coreference-related aspects:

- large-scale annotation of near-identity introduced by [3], cf. Section 4, which obviously could not be performed by approaches published before 2010, and seems not to have been applied since then except in our work;
- dominant expressions (cf. Section 5), whose idea appears in none of the studied approaches, despite its utility e.g. for cross-document coreference annotation;
- semantic heads (cf. Section 6), whose difference from syntactic heads does not seem to be an issue in other languages than Polish, for which coreference annotation has been performed.

As for taking zero subjects into account, the five major approaches concerned are [1] (for Bulgarian), [3] (for Spanish), [10] (for Japanese), [12] (for Arabic and Chinese) and ours (for Polish). In Section 7 we revisit this notion in order to report on its nature and frequency in our corpus.

4 Near-Identity

Near-identity is a novel coreference relation defined in [14]. Our understanding of this concept, as discussed in [20] includes two phenomena: (i) two mentions refer to

² See http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/co_task.html.

the *same* entity but the text suggests the opposite (refocusing, e.g. *pre-war Warsaw* vs. *today's Warsaw*), (ii) two mentions refer to *different* entities but the text suggests the opposite (neutralization, e.g. *wine* as a bottle vs. its contents). [15] state that the binary distinction between coreference (identity) and non-coreference (non-identity) is too limited, since a continuum of values exists between these two extreme cases. Near-identity should help in bridging this gap. The same paper puts forward a fine-grained typology of near-identity with 4 types (name metonymy, meronymy, class, spatio-temporal function) and 15 subtypes (role, location, organization, etc.).

While the idea of near-identity itself was inspiring, the applicability of its typology seemed uncertain. Thus, before including it in our annotation schema we first studied the reliability of detecting near-identity alone, i.e. regardless of its type. As discussed in Section 9, the value obtained for inter-annotator agreement in untyped near-identity links, in terms of Cohen's κ , was only 0.222. Some annotators never even used the near-identity links, which proves that the concept is hard to capture.

These results bring strong doubts not only about the utility of the mentioned typology but also of the near-identity as such. The concept of near-identity might be, in our opinion, a result of mixing two different levels of language: the meaning of a word and its reference. The former is independent of the context while the latter is a function of a word used in a given context. Words very often have common elements of meaning – that is why it was possible to create semantic networks like WordNet (web of words which are linked with each other without any context). Reference though is related more to pragmatics than to semantics. Very often phrases formed with words sharing no semantic elements can refer to the same referent (e.g.: *football players of Polonia* and “*Black shirts*”), and the link between such phrases can be established only due to the external knowledge (Polonia football players wear black shirts). On the other hand the same words can refer to different referents, e.g.:

- (5) *Te tipsy bardzo niszczą paznokcie [...] ostatnio właśnie już mi całe paznokcie odrosły już nawet już nie mam takiej strasznie zniszczonej płytki po tych paznokciach.*

'These artificial nails damage nails a lot [...] lately my nails have just grown back I don't have so awful haggard nail any more after those nails.'

In this text the three occurrences of *nails* have different referents (although still the same basic semantics): generic nails, nails of the speaker, and artificial nails.

Our experience with the corpus annotation shows that people usually have no problem with distinguishing these two linguistic levels: the word meaning and the word reference. Conversely, near-identity links seem rather hard to establish and no repeatable pattern in the near-identity annotation has occurred. Therefore in our opinion the utility of the near-identity concept for coreference annotation is questionable.

5 Dominant Expressions

In every cluster we indicate the *dominant expression*, i.e. the expression that carries the richest semantics or describes the referent the most precisely. The best candidates for dominant expressions are named entities, as well as periphrastic phrases that denote a particular object in the discourse world, e.g:

- (6) Cluster: *David Beckham, rozgrywający Realu Madryt* ‘David Beckham, Real Madryt player’ Dominant expr.: *David Beckham*

In many cases, pointing at the dominant expression helps the annotators sort out a large set of pronouns denoting various persons (e.g. in fragments of plays or novels). We think that it might also facilitate cross-document annotation or the creation of a semantics frame containing different descriptions of the same object.

In 62% of all cases, the dominant expression was selected from among NGs contained in the cluster. 77% of them were taken without any changes (which means that there was the base form of the NG in the cluster) as in (7), while 23% of them were transformed into their base forms.

- (7) Cluster: *tamtejszy dziennikarz, dziennikarz, Ja, napisał, pismak*
 ‘local journalist, journalist, I, wrote, hack.’
 Dominant expr.: *tamtejszy dziennikarz*
 ‘local journalist’.

For 38% of the clusters, the dominant expression was not present in the text but given by the annotator instead (e.g., *Halloween* for the cluster containing a repeated phrase: *tej okazji* ‘this occasion’). This was necessary in particular when the cluster consisted of verb forms only, e.g.:

- (8) Cluster: *stwierdzili, powiedzieli* ‘stated, said’
 Dominant expr.: *lekarze w Polsce* ‘doctors in Poland’

As mentioned in Section 9, dominant expressions can be annotated with a much higher reliability (66.78%) than near-identity. A detailed study of disagreement cases shows that many of them are superficial rather than essential. They are due e.g. to different letter case or spelling errors in the dominant expressions, or to the fact that some annotators produce the base form of the dominant expressions while others cite the (inflected) forms occurring in the corpus. Such cases may be corrected mostly automatically, which will enhance the inter-annotator agreement indicator.

6 Semantic Heads

For each mention, its semantic head is selected, being the most relevant word of the group in terms of meaning. The semantic head of typical nominal group is the same element as the syntactic head but in numeral groups the numeral is the syntactic head, and the noun is the semantic head. Numeral groups are regarded as nominal groups in our project (e.g., *dużo pieniędzy* ‘a lot of money’, *trzech z was* ‘three of you’). They can be also embedded in other nominal groups (e.g., *sąsiad dwóch kobiet* ‘neighbour of two women’). In these examples, words *dużo, trzech, dwóch* ‘a lot, three, two’ are syntactic heads and *pieniędzy, was, kobiety* ‘money, you, women’ are semantic heads.

The reason why we are interested in semantic rather than in syntactic heads is the same as when we admit a very broad definition of nominal groups (including numeral

phrases, some adjectival phrases, etc., cf. Section 2). Namely, coreference is a phenomenon on the level of semantics and discourse more than syntax. Thus, understanding the semantically central elements should help establish discourse links, notably in future automatic coreference resolvers. In particular, it seems promising to examine agreement in case, gender, number, synset, etc. between semantic heads in potentially co-referring mentions.

As shown in Section 9, the reliability of annotating the semantic heads was very high (97.00%). Disagreement resulted mainly from inattention in distinguishing syntactic from semantic heads, e.g., a) adjectives or numerals were selected instead of nouns, b) the head of a subordinate phrase was selected instead of the head of the main phrase (e.g., *metropolii* ‘metropolis’ was marked as the semantic head in: *niedobrą dzielnicę jakiejś wieloetnicznej metropolii* ‘bad quarter of a multi-ethnic metropolis’).

7 Zero Subjects

Similarly to most other Slavic languages, Polish grammar permits independent clauses to lack explicit subjects. The form of the null referent is then partially indicated by the morphology of the verb. We annotate such cases with identity links, as in (9) while other types of elliptic expressions are not linked, unlike in [10], cf. (10)–(11).

(9) *Maria wróciła już z Francji. \emptyset Spędziła tam miesiąc.*

‘Maria wróciła już z Francji. \emptyset Spędziła tam miesiąc.’

(10) *Janek kupił duże pudełko czekoladek, ale niewiele \emptyset już zostało.*

‘John bought a huge box of chocolates, but there were just a few \emptyset left.’

(11) *Czytałeś książki Lema? Czytałem \emptyset .*

‘Have you read Lem’s books? I have \emptyset .’

Even with such limitation, the phenomenon seems frequent — there are 4678 coreference clusters containing at least one zero subject (26.89% of the total number of non-singleton clusters).

8 Pronominal Coreference and Other Issues

Originally, we had excluded some types of pronouns (indefinite, negative, reflexive, interrogative) from the annotation on the assumption of their non-referentiality. Surprisingly, the analysis of the corpus showed that they can frequently form coreferential chains. Very often an indefinite pronoun is a subject of a verb sequence (in which the second verb is a zero subject verb), e.g.:

(12) *Jak ktoś jest zazdrosny, znaczy, że naprawdę \emptyset kocha.*

‘If someone is jealous, it means, that (he/she) really loves.’

Sometimes pronouns make a typical anaphoric link: a demonstrative (pronoun) refers to indefinite pronoun used in the first clause, cf.:

- (13) *Jeśli coś przestanie być potrzebne, można to usunąć z dysku, zwalniając miejsce na inne zasoby.*

'If something is no longer needed, one can remove it from the disk to save on storage for other resources.'

Indefinite pronouns can also implicitly refer to a specific person. The speaker in the example below does not want to speak openly about the former director and uses the indefinite pronoun *kogoś* 'someone' instead:

- (14) *Po rezygnacji z pracy w szpitalu były dyrektor zniknął z życia publicznego. ØWrócił dopiero, gdy starosta Andrzej Barański zaproponował mu współpracę. Posunięcie starosty, wywołało ostrą reakcję kilku radnych. W trakcie ostatniej sesji kilkakrotnie pytano, czy nowy pracownik ma odpowiednie kwalifikacje, by zdobywać dla powiatu unijne środki pomocowe. – Uważam, że powołanie kogoś, kto nie sprawdził się w szpitalu i jako szef spółdzielni mieszkaniowej, może budzić wątpliwości – mówi Wojciech Wenecki.*

'After giving up the job in the hospital, the former director had disappeared from the public life. (He) came back only when starost Andrzej Barański offered him cooperation. The move of the starost provoked sharp reaction of several councillors. During the last session they repeatedly asked if the new employee has suitable qualifications to obtain for the local government administration UE aid resources. - I think that appointing someone who haven't performed well in the hospital and as chief of the housing cooperative may raise doubts - says Wojciech Wenecki.'

The examples of coreferential chains containing indefinite pronouns show that these pronouns should have been allowed in coreference chains. We wish to reconsider this phenomenon at the end of the annotation process. It makes us think that the problem of coreference in the text might be somehow different than the one of the reference to the world. Maybe it should be examined as a separate phenomenon.

9 Inter-Annotator Agreement

For the purpose of measuring the inter-annotator agreement, a sample of the corpus (henceforth called the IAA-sample) was annotated independently by two annotators. More precisely, several annotators participated in the experiment but each text was annotated by exactly two of them. The IAA-sample consisted of 210 texts selected so as to uniformly represent the 14 existing text genres (prose, press, dialogues, etc.). It contained 60674 tokens in total, i.e. about 12% of the whole corpus.

The establishment of the inter-annotator agreement with respect to mention detection remains a challenge [5]. Thus, we based our estimation on the F-measure (which does not take agreement by chance into account) — it amounted to 85.55% in the IAA-sample. Namely, the two annotators produced 20420 and 20560 mentions, respectively, and 17530 of them had identical borders in both sets (including the internal borders in case of discontinuous mentions). Further calculations, resumed in Table 1, take only those 17530 mentions into account whose borders were marked identically by both annotators.

Chance-corrected agreement of near-identity links with Cohen's κ [16] was calculated for each text separately (because cross-document links are not allowed) and then averaged. For a single text the agreement was measured for the decisions on every pair

Table 1. Detailed inter-annotator agreement

Mentions	Near-identity	Dominant expressions	Semantic heads	Identity clusters
$F_1 = 85.55\%$	$\kappa = 0.222$	66.78%	$S = 97.00\%$	$\alpha = 79.08\%$

of mentions (marked by both annotators) in the text – whether they were marked as near-identical or not. The probability of the annotator marking two mentions as near-identical was estimated for each text and annotator separately. If there were no near-identity links in a text its agreements was equal to 1. If however one annotator marked some near-identity links in a text and the other annotator marked no such link in the same text, the agreement for this text was 0 (because the expected agreement was the same as the observed agreement). The final result is very low – only 0.222. Frequently (128 times) a link was marked as near-identity by one annotator and as identity by the other. These figures prove the difficulty of reliably annotating near-identity links.

The agreement in annotating dominant expressions was calculated only on those non-singleton clusters which were identically designated by both annotators (among the identically delimited mentions). There were 6162 such clusters, out of which 4115 (about 66,78%) had the same dominant expression. If we count such proportion for only one mention from each cluster (as each cluster member has the same dominant expression), for 1818 such cluster "representatives" 1146 mentions (63,04%) have the same dominant expression in both annotations. No chance-corrected analysis was conducted because the dominant expressions could have been entered by the annotators as free text, which jeopardizes the probability model of agreement by chance.

The agreement in annotating semantic heads was evaluated in terms of the chance-corrected agreement (for identically delimited mentions). Uniform probability distribution among all possible head choices was assumed and the S measure [17] was used. The observed agreement and the expected agreement were close to 99.05% and 68.32%, respectively. The former was quite high mainly because of the large number of mentions consisting of only one token (therefore only one possible head). These two values yield the final S result of approximately 97.00%.

For completeness, we also present the chance-corrected inter-annotator agreement of the identity clustering task, calculated according to the version of weighted Krippendorff's α [18] proposed by Passoneau in [19].

10 Conclusions and Perspectives

We have presented a detailed study of several particularly interesting phenomena related to coreference annotation in an inflectionally rich language such as Polish. We have applied the novel concept of near-identity [14] on a large scale and we came to the conclusion that it cannot be reliably applied in coreference annotation. We also argue that semantic heads are more relevant to coreference than syntactic heads. We have introduced a novel idea of dominant expressions which represent the expressions carrying the biggest semantic load with respect to a reference cluster. These may be useful e.g. in cross-document coreference annotation. Finally, we have reviewed our previous conviction that indefinite, negative and other particular types of pronouns can never appear in

coreference chains. We hope that all these observations may contribute to a high-quality methodology and usefulness of future coreference annotation projects, particularly in highly inflected languages.

References

1. Osenova, P., Simov, K.: BTB-TR05: BulTreeBank Stylebook. BulTreeBank Version 1.0. Technical Report BTB-TR05, Linguistic Modelling Laboratory, Bulgarian Academy of Sciences, Sofia, Bulgaria (2004)
2. Nedoluzhko, A., Mírovský, J., Ocelák, R., Pergler, J.: Extended Coreferential Relations and Bridging Anaphora in the Prague Dependency Treebank. In: Proceedings of the 7th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2009), Goa, India, pp. 1–16. AU-KBC Research Centre, Anna University, Chennai (2009)
3. Recasens, M., Martí, M.A.: AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan. *Language Resources and Evaluation* 44(4), 315–345 (2010)
4. Korzen, I., Buch-Kromann, M.: Anaphoric relations in the Copenhagen Dependency Treebanks. In: Proceedings of DGfS Workshop, Göttingen, Germany, pp. 83–98 (2011)
5. Poesio, M., Artstein, R.: Anaphoric Annotation in the ARRAU Corpus. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco, pp. 1170–1174. European Language Resources Association (2008)
6. Recasens, M.: Coreference: Theory, Annotation, Resolution and Evaluation. PhD thesis, Department of Linguistics, University of Barcelona, Barcelona, Spain (2010)
7. Linguistic Data Consortium: ACE (Automatic Content Extraction) Spanish Annotation Guidelines for Entities (2006), http://projects.ldc.upenn.edu/ace/docs/Spanish-Entities-Guidelines_v1.6.pdf (accessed on February 18, 2013)
8. Hinrichs, E.W., Kübler, S., Naumann, K.: A Unified Representation for Morphological, Syntactic, Semantic, and Referential Annotations. In: Proceedings of the ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky, Ann Arbor, Michigan, USA, pp. 13–20 (2005)
9. Magnini, B., Pianta, E., Girardi, C., Negri, M., Romano, L., Speranza, M., Lenzi, V.B., Sprugnoli, R.: I-CAB: the Italian Content Annotation Bank. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), Genova, Italy, pp. 963–968. European Language Resources Association (2006)
10. Iida, R., Komachi, M., Inui, K., Matsumoto, Y.: Annotating a Japanese Text Corpus with Predicate-Argument and Coreference Relations. In: Proceedings of the Linguistic Annotation Workshop (LAW 2007), pp. 132–139. Association for Computational Linguistics, Stroudsburg (2007)
11. Pradhan, S.S., Ramshaw, L., Weischedel, R., MacBride, J., Micciulla, L.: Unrestricted Coreference: Identifying Entities and Events in OntoNotes. In: Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007), pp. 446–453. IEEE Computer Society, Washington, DC (2007)
12. Weischedel, R., Pradhan, S., Ramshaw, L., Kaufman, J., Franchini, M., El-Bachouti, M.: OntoNotes Release 4.0 (2010), <http://www.bbn.com/NLP/OntoNotes> (accessed on February 18, 2013)
13. Hendrickx, I., Bouma, G., Daelemans, W., Hoste, V., Kloosterman, G., Mineur, A.M., Van Der Vloet, J., Verschelde, J.L.: A Coreference Corpus and Resolution System for Dutch. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco, pp. 144–149. European Language Resources Association, ELRA (2008)

14. Recasens, M., Hovy, E., Martí, M.A.: Identity, non-identity, and near-identity: Addressing the complexity of coreference. *Lingua* 121(6) (2011)
15. Recasens, M., Hovy, E., Martí, M.A.: A Typology of Near-Identity Relations for Coreference (NIDENT). In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, Valletta, Malta, pp. 149–156. European Language Resources Association (2010)
16. Cohen, J.: A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20(1), 37–46 (1960)
17. Bennet, E.M., Alpert, R., Goldstein, A.C.: Communications through limited response questioning. *Public Opinion Quarterly* 18, 303–308 (1954)
18. Krippendorff, K.H.: *Content Analysis: An Introduction to Its Methodology*, 2nd edn. Sage Publications, Inc. (December 2003)
19. Passonneau, R.J.: Computing reliability for coreference annotation. In: *LREC*. European Language Resources Association (2004)
20. Ogrodniczuk, M., Głowińska, K., Kopeć, M., Savary, A., Zawisławska, M.: Interesting Linguistic Features in Coreference Annotation of an Inflectional Language. In: Sun, M., Liu, T., Sun, L., Zhang, M., Sun, M., Lin, D., Wang, H. (eds.) *CCL and NLP-NABD 2013*. LNCS (LNAI), vol. 8202, pp. 97–108. Springer, Heidelberg (2013)

Semi-supervised Learning with Transfer Learning

Huiwei Zhou, Yan Zhang, Degen Huang, and Lishuang Li

Dalian University of Technology, Dalian, Liaoning, China
{zhouhuiwei, huangdg}@dlut.edu.cn, 358742700@mail.dlut.edu.cn,
lilishuang314@163.com

Abstract. Traditional machine learning works well under the assumption that the training data and test data are in the same distribution. However, in many real-world applications, this assumption does not hold. The research of knowledge transfer has received considerable interest recently in Natural Language Processing to improve the domain adaptation of machine learning. In this paper, we present a novel transfer learning framework called TPTSVM (Transfer Progressive Transductive Support Vector Machine), which combines transfer learning and semi-supervised learning. TPTSVM makes use of the limited labeled data in target domain to leverage a large amount of labeled data in source domain and queries the most confident instances in target domain. Experiments on two data sets show that TPTSVM algorithm always improves the classification performance compared to other state-of-the-art transfer learning approaches or semi-supervised approaches. Furthermore, our algorithm could be extended to multiple source domains easily.

1 Introduction

Traditional machine learning assumes that the training data and test data are in the same distribution and the training data is sufficient to get an acceptable model [1][2][3]. But in many real works, the training data are always scarce and it is expensive to label the sufficient training data. For example, in Web-document classification, the labeled data used for training may be easily outdated or under a different distribution from the new data [4][5][6]. In such cases, it would be helpful if we could use unlabeled new data or transfer the classification knowledge into the new domain.

Transfer learning is a machine learning algorithm that allows the distributions, domains, even tasks used in training data and testing data differently [7][8]. Recently, research on transfer learning has attracted more and more interest in several topics, such as knowledge transfer [9], learning to learn [10], multi-task learning [11], domain transfer [12]. Among these, we address on the domain transfer problem, which aims to transfer classification knowledge from the source domain to the target domain in the same task.

Domain-transfer learning has been studied and works well when the distributions of source and target domain are similar [13][14]. But significant distribution divergence might cause negative transfer [15], which is the limitation of transfer learning. Finding the inner relationship between source domain and target domain and using the source data as far as possible are the problems must be solved. TrAdaBoost [16]

decrease the negative effects of source domain and boost the accuracy on target domain by Boosting. TransferBoost [17] adjusts the weights of each source domain according to its positive transferability to the target domain. Active Vector Rotation (AVR) [18] uses active learning to avoid negative transfer and reduces the labeling cost. Most current transfer learning researches focus on using a large set of labeled source data and a small set of labeled target data [19][20]. However, the limited labeled target data can not represent the whole target domain sufficiently.

On the other hand, semi-supervised learning is another effective machine learning strategy when the training sets are small, for example, in sentiment analysis [21]. Including a particular set of unlabeled working or testing data, semi-supervised learning can be used to improve the generalization performance for both training and working data set. Transductive Support Vector Machines (TSVM) [22] is a well known semi-supervised learning algorithm, which achieves better performance than traditional inductive SVM, especially for small training sets. However, TSVM assumes that the training and working data follow the same ratio of positive/negative examples. To handle different class distribution, progressive transductive support vector machine (PTSVM) [23] labels and modifies the unlabeled working data set with the rule of pairwise labeling and dynamical adjusting. Semi-supervised learning only uses a small number of labeled training data and a large number of unlabeled testing data. A large number of labeled data from a similar old domain often provide some useful information. Throwing away all old data would also result in a waste.

In this paper, we propose a framework, named transfer progressive transductive support vector machine (TPTSVM), which takes advantage of both transfer learning and semi-supervised learning techniques. TPTSVM tries to make use of the limited labeled data in target domain to leverage a large amount of labeled data in source domain and queries the most confident instances in target domain. The weights of individual instances are carefully adjusted on both the instance level and the domain level. In our experiments, we only use one source domain, but our algorithm could be extended to multiple source domains easily. The experiments show that semi-supervised learning can improve the transfer learning's performance and our algorithm works well especially for insufficient training set.

2 Transfer Progressive Transductive Support Vector Machine

To simplify the question, we constrain the discussion of TPTSVM to binary classification tasks and transfer knowledge from one source domain to the target domain. But in fact, TPTSVM can also be extended to multi-class and multi-source. Given three data sets, the target domain labeled data set $D_l = \{(x_i^l, y_i^l) | i = 1, \dots, n\}$, $y_i^l = \{-1, +1\}$, the target domain unlabeled data set $D_u = \{(x_j^u) | j = 1, \dots, m\}$, and the source domain labeled data set $D_s = \{(x_k^s, y_k^s) | k = 1, \dots, r\}$, $y_k^s = \{-1, +1\}$. n , m and r are the sizes of D_l , D_u and D_s . Assume that D_l and D_u are in the same domain with same distribution, but D_l is not sufficient to training a model to classify the D_u . Our TPTSVM algorithm tries to use both D_s and D_u to help the insufficient training set D_l to train a better

Algorithm 1

Input the two labeled data sets D_s and D_l , the unlabeled data set D_u , a SVM learner $SVM(D(x_i), W(x_i))$, the number of iteration N , the growth size p at each iteration ($p/2$ positive and $p/2$ negative confident instances).

Initialize

- a) $D_{ul} = \{\}, D_{uu} = D_u$.
 b) $W^1 = (w(x_i)) = 1, x_i \in D_s \cup D_l \cup D_u$,
 $W_l \in W = (w(x_i)), x_i \in D_l \cup D_u$

$$c) \beta = \frac{1}{1 + \sqrt{2 \ln |D_s| / N}}$$

For $d=1, 2, \dots, N$

1. Call learner $SVM(D_s \cup D_l \cup D_{ul}, W^d)$, get back a hypothesis function $f(x_i)$;

Call learner $SVM(D_l \cup D_u, W_l^d)$, get back a hypothesis function $f_l(x_i)$.

2. Calculate the error of $f(x_i)$ on D_l :

$$\varepsilon = \sum_{x_i \in D_l} \frac{W_l(x_i) * |\text{sign}(f(x_i)) - y_i|}{2 * \sum_{x_i \in D_l} W_l^d(x_i)}$$

Calculate the error of $f(x_i)$ on $D_l \cup D_u$:

$$\varepsilon_s = \sum_{x_i \in D_l \cup D_u} \frac{W_l(x_i) * |\text{sign}(f(x_i)) - y_i|}{2 * \sum_{x_i \in D_l \cup D_u} W_l^d(x_i)}$$

Calculate the error of $f_l(x_i)$ on $D_l \cup D_u$:

$$\varepsilon_l = \sum_{x_i \in D_l \cup D_u} \frac{W_l(x_i) * |\text{sign}(f_l(x_i)) - y_i|}{2 * \sum_{x_i \in D_l \cup D_u} W_l^d(x_i)}$$

3. Reweight the instances x_i on D_s :

$$w^{d+1}(x_i) = w^d(x_i) * e^{(\varepsilon_l - \varepsilon_s)} * \beta^{|\text{sign}(f(x_i)) - y_i|/2}, x_i \in D_s;$$

Reweight the instances x_i on D_l :

$$w^{d+1}(x_i) = w^d(x_i) * (1 - \varepsilon) / \varepsilon, \text{ if } \text{sign}(f(x_i)) \neq y_i, x_i \in D_l.$$

4. Calculate the function values $f(x_i)$ of instances x_i on D_{uu} and D_{ul} .
 5. Select $p/2$ positive and $p/2$ negative the most confident instances from D_{uu} to D_{ul} .
 6. Move unconfident instances from D_{ul} back to D_{uu} .
 7. Reweight instances x_i on D_{ul} :

$$w^{d+1}(x_i) = \frac{n}{N} * (1 - \varepsilon), x_i \in D_{ul}.$$

Output the hypothesis function $f(x_i)$.

classifier $f(x_i)$ that minimizes the prediction error on the target domain unlabeled data set D_u . A formal description of the framework is given in Algorithm 1. Where, D_l is not sufficient to train a model to classify the D_u ; D_s is the old data set or source domain data set that we try to reuse as much as possible. TPTSVM tries to transfer source instances to learn the target distribution on both the instance level and the domain level. On the domain level, TransferBoost is used to adjust the weight of source domain to overcome the irrelevancies of source domain. On the instance level, the theory of Adaboost is applied to adjust the weights of instances.

Besides D_l and D_s , the learner is also given the unlabeled data D_u from the target domain. D_u is large enough to response the feature and the distribution of the target domain data. We also expect to make use of D_u for target-domain classification. TPTSVM aims to select the most confident instances from D_u into the predicted labeled data set D_{ul} to help learning. Semi-supervised algorithm is a learning framework which aims to label the unlabeled data for training. In our TPTSVM, the improved semi-supervised learning algorithm PTSVM is applied to select the most informative data.

On each iteration round, two models are trained with instances' weights. One uses the union of the source and the target domain data including the predicted labeled data set D_{ul} selected from the unlabeled data set D_u , and the other just uses the training data set in the target domain. TPTSVM trains two classifiers with two sets of training instances together with their weights. It then reweights each instance in the source domain, increasing or decreasing their weights by a factor of $e^{(\epsilon_i - \epsilon_s)}$ based on whether the union of the source and target domain data shows positive or negative transfer to the target domain, and reducing their weights by a factor of $\beta^{|sign(f(x_i)) - y_i|/2}$ if a source domain instance is mistakenly predicted. The parameter β in algorithm hedge(β) [24] is used to decrease the weight of harmful instances. $e^{(\epsilon_i - \epsilon_s)}$ is an influential factor of source domain data. It will be less than 1 when the source domain is irrelevant and larger than 1 when relevant, which makes TPTSVM easy to extend to multiple source domains. To boost the accuracy on the target training data D_t , TPTSVM increases the weights of mispredicted instances using the error rate ϵ computed from D_t .

Next, TPTSVM selects $p/2$ positive and $p/2$ negative instances, which are the most confidently predicted and useful (the hypothesis function value $0.5 < |f(x_i)| < 1$), from D_{ul} to D_{ul} . Similarly, it moves unconfident instances ($|f(x_i)| < 0.5$) from D_{ul} back to D_{uu} . The weights of instances in D_{ul} are increased slightly on each iteration based on the training error \mathcal{E} on the target domain.

After several iterations, the weights of the source domain instances will be changed according to the similarity of distribution between the source data and the target data. In addition, the instances from the source domain that show "good"("bad") effect to the target domain will have higher (lower) training weights, while target instances that are mislabeled will be emphasized. Furthermore, the credible predicted unlabeled target-domain data are available and used to represent the target distribution. In this manner, TPTSVM trains a domain adaptation model from both source and target data via transfer learning and transductive learning.

3 Analysis

In general SVM [22] framework where we cannot separate training instances linearly, by introducing slack variables ξ_i , the primal problem can be written as:

$$\begin{aligned} \min & \left(\frac{1}{2} \|w\|^2 + C \sum_i \xi_i \right), \\ \text{s.t.} \quad & y_i (w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \end{aligned} \quad (1)$$

It has been proved that the unlabeled data inside the margin band of the separating hyperplane could improve the performance in semi-supervised machine learning. By including some unlabeled data, TSVM [22] becomes solving the following optimization problem:

$$\begin{aligned} \text{minimize over } & (y_1^*, \dots, y_m^*, w, b, \xi_1, \dots, \xi_n, \xi_1^*, \dots, \xi_m^*) \\ & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i + C^* \sum_{j=1}^m \xi_j^* \\ \text{subject to : } & \forall_{i=1}^n : y_i [w \cdot x_i + b] \geq 1 - \xi_i \\ & \forall_{j=1}^m : y_j^* [w \cdot x_j^* + b] \geq 1 - \xi_j^* \\ & \forall_{i=1}^n : \xi_i \geq 0 \\ & \forall_{j=1}^m : \xi_j^* \geq 0 \end{aligned} \quad (2)$$

Where y_1^*, \dots, y_m^* are the predicted labels for the instances in D_u . In our TPTSVM algorithm, not only the target domain unlabeled data, but also the source domain labeled data is added in training set. So the optimization problem becomes:

$$\begin{aligned} \text{minimize over } & (y_1^*, \dots, y_m^*, w, b, \xi_1, \dots, \xi_n, \xi_1^*, \dots, \xi_m^*) \\ & \frac{1}{2} \|w\|^2 + \sum_{k=1}^r C_k' \xi_k + \sum_{i=1}^n C_i \xi_i + \sum_{j=1}^m C_j^* \xi_j^* \\ \text{subject to : } & \forall_{k=1}^r : y_k [w \cdot x_k + b] \geq 1 - \xi_k \\ & \forall_{i=1}^n : y_i [w \cdot x_i + b] \geq 1 - \xi_i \\ & \forall_{j=1}^m : y_j^* [w \cdot x_j^* + b] \geq 1 - \xi_j^* \\ & \forall_{k=1}^r : \xi_k \geq 0 \\ & \forall_{i=1}^n : \xi_i \geq 0 \\ & \forall_{j=1}^m : \xi_j^* \geq 0 \end{aligned} \quad (3)$$

The objective function (3) uses three data sets for learning: source domain labeled data, target domain labeled data and target domain unlabeled data. C_k , C_i and C_j^* are the loss costs of instances in source domain labeled data, target domain labeled data and target domain unlabeled data respectively. The final optimal separating hyperplane could be found after finite iterations because the loss costs are finite numbers.

To reduce irrelevance of the source domain data, the algorithm Hedge(β) [24] is used in our algorithm. We have the same conclusion with the algorithm Hedge(β):

$$\frac{L_t}{N} \leq \min_{1 \leq j \leq r} \frac{L(x_j)}{N} + \sqrt{\frac{2 \ln r}{N}} + \frac{\ln r}{N} \tag{4}$$

It indicates that the average training loss (L_t/N) through N iteration on the source domain data D_t is at most $\sqrt{2 \ln r / N} + \ln r / N$ larger than the average minimum training loss of the instances ($\min_{1 \leq j \leq r} L(x_j) / N$).

Like algorithm Adaboost [24], our algorithm TPTSVM increases the weights of mispredicted instances in target domain. Therefore, the accuracy on target domain labeled data which is believed as the standard data could be guaranteed.

In algorithm Adaboost, the prediction error ϵ_p on target domain labeled data satisfies the follow formula:

$$\epsilon_p \leq 2^N \prod_{d=1}^N \sqrt{\epsilon_d (1 - \epsilon_d)} \tag{5}$$

if the final hypothesis is:

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{d=1}^N (\log \frac{1}{\beta_d}) h_d(x) \geq \frac{1}{2} \sum_{d=1}^N \log \frac{1}{\beta_d} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

When ϵ_d is less than 1/2, the prediction error of $h_f(x)$ on the training data becomes increasingly smaller after each iteration.

Formally, if we consider the follow hypothesis

$$h_f^*(x) = \begin{cases} 1 & \text{if } \sum_{d=\tau}^N (\log \frac{1}{\beta_d}) h_d(x) = \sum_{d=\tau}^N \log \frac{1}{\beta_d} \\ 0 & \text{if } \sum_{d=\tau}^N (\log \frac{1}{\beta_d}) h_d(x) = 0 \end{cases} \tag{7}$$

we have:

$$\epsilon_p \leq 2^{N-\tau+1} \prod_{d=\tau}^N \epsilon_d \tag{8}$$

We assume the distribution of the target domain unlabeled data is the same as that of the target domain labeled data. In our algorithm TPTSVM, an unlabeled instance x_τ which is added to D_{ul} after the τ^{th} iteration will still be in D_{ul} after all iteration. x_τ would satisfies the hypothesis $h_f^*(x)$, and the error probability of x_τ would less than $2^{-N-\tau+1} \prod_{d=\tau}^N \epsilon_d$. Actually, the error probability of the instances added into D_{ul} earlier will become smaller along with the $N - \tau + 1$ growing bigger. On the other hand, the accuracy of the instances added into D_{ul} later is also acceptable, because we increased the weights of mispredicted instances in target domain.

4 Experiment

4.1 Data Sets

The experiments are performed on one none-text data set (mushroom data from the UCI machine learning repository¹) and one text data set (20 newsgroups data²), which are all used in [16]. We also split the data sets to fit our learning problem like [16]. We generate four tasks using the two data sets as shown in Table 1.

The mushroom data contains two categories, edible and poisonous. We split the data based on the stalk-shape feature to make the source and target data have different distribution. The source data set consists of all the instances whose stalks are enlarging, while the target data set consists of the instances whose stalks are tapering.

For 20newsgroups data (Task2, Task3 and Task4) contains seven top categories and 20 subcategories. The task is defined as the top-category-classification problems. We split the data based on the subcategories. For example, in Task2, we learn to classify two classes, sci and talk. The target data set consists of two subcategories sci.space and talk.religion.misc, while the source data set consists of all other subcategories data under the top categories sci and talk, i.e. sci.crypt, sci.electronic, sci.med, talk.politics.guns, talk.politics.mideast and talk.politics.misc. Task3 and Task4 are similar to Task2.

Table 1. Data sets description

Task	Data set	Size	
		$D_t \cup D_u$	D_s
1	edibles vs poisonous	4608	3156
2	sci.space vs talk.religion.misc	4880	2315
3	rec.sport.hockey vs sci.space	5089	2537
4	rec.sport.hockey vs talk.religion.misc	4883	2320

¹ <http://archive.ics.uci.edu/ml/datasets/Mushroom>

² <http://qwone.com/~jason/20Newsgroups>

4.2 Comparison Methods

We compare our algorithm with four algorithms: SVM trained on only the target training data D_t , SVMt and TrAdaBoost trained on both target and source training data D_s , and PTSVM trained on both target labeled data D_t and unlabeled data D_u .

SVM^{light} [25] is used as the basic learner in our experiments. When training SVM, the training data set only consists of the target domain labeled data which is not enough to train a good classifier. Except for target domain labeled data, SVMt also uses source domain labeled data D_s as additional training set. Besides, SVMt is the same as SVM. So the result of SVMt reflects the effect of source data. The training set used in TrAdaBoost is the same as SVMt, but it adjusts the weights of all training instances through iterations. So the benefit of TrAdaBoost is brought by transfer learning compare to SVMt. PTSVM selects the instances from the unlabeled target data set D_u , and adds them to D_t .

Our TPTSVM algorithm is trained on the three data sets: D_t , D_s and D_u , trying to improve the transfer learning with semi-supervised learning.

We also tried to experiment with two other methods, Frustratingly Easy Semi-Supervised Domain Adaptation [26] and TrAdaBoost employing TSVM, which also training with unlabeled data, but the results was not satisfied on our data sets.

4.3 Experiment Results

The parameters of SVM are set to the default values of SVM^{light}. Iteration time N of both TrAdaBoost and TPTSVM is 100. The growth size p at each iteration in TPTSVM is set to 20. All the results below are the average of 10 repeats by random.

The results on the mushroom data are showed in Figure 1. To investigate the effect of the target training size to each algorithm, we use 50, 100 and 150 target instances as training set respectively (positive and negative instances are half and half). From the results we can see, when the training data is not sufficient (50 instances), the performance of SVM is poor. The knowledge from the source data may help the SVMt learner. While when training instance number is 150, it is enough to train a good SVM classifier. The additional training set of source domain takes negative effect to the SVMt learner. Therefore, the source training data contain not only good knowledge, but also noisy data. TrAdaBoost could always perform better than SVMt and

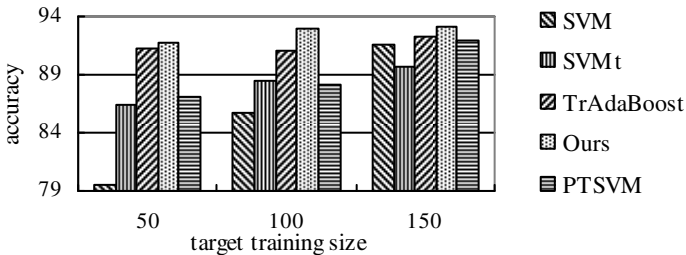


Fig. 1. Accuracy on mushroom data

SVM, because it could reduce the effect of the noisy source data, while transfer useful knowledge to the target domain. PTSVM also improves the accuracy of SVM by using the target unlabeled data to help learning. Our algorithm gives the best performance steadily through the novel combination of transfer and transductive learning.

As for Task 2, figure 2 compares the test classification accuracy of the algorithms when training on the different size of the target training set. It can be seen that the results with different training size show similar trend as Task1 on mushroom data. When the number of training set is 50, TrAdaBoost performs worse than SVMt. But we found that the accuracy on target domain training set is 100%. This may be caused by the over-fitting of AdaBoost. Our algorithm also uses the boosting of AdaBoost, but our algorithm could effectively overcome the over-fitting of AdaBoost by semi-supervised learning. Figure 3 shows the accuracy curves of TrAdaBoost and our algorithm with different numbers of iterations. The size of the target training set is 400. From the curves, we can see how semi-supervised learning helps to improve transfer learning in normal situation.

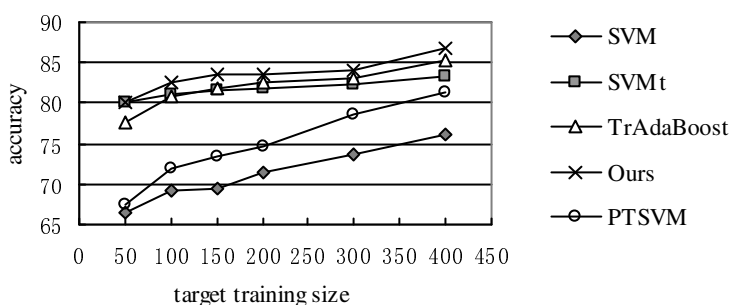


Fig. 2. Accuracy on Task2

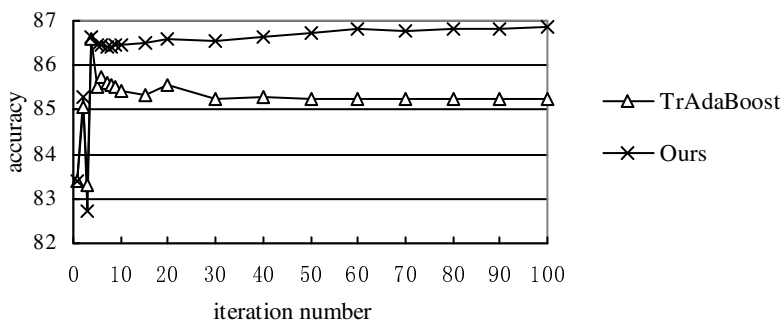


Fig. 3. Learning curves on Task2

Table 2 shows comparison results on test classification tasks (Task2, Task3 and Task4). We do the same experiment on them. The number of training instances is 150. Seen from the table, our algorithm outperforms all four algorithms over all test classification tasks.

Table 2. Comparison results on test classification tasks

Algorithm	Task 2	Task 3	Task 4
SVM	69.42	56.17	68.83
SVMt	81.58	62.20	79.40
TrAdaBoost	81.84	63.25	80.52
PTSVM	73.40	57.35	66.36
Ours	83.47	63.86	81.25

5 Conclusion

TPTSVM is a novel approach to knowledge transfer and domain adaptation by combining transfer learning and semi-supervised learning. The proposed TPTSVM algorithm transfers the source knowledge to the target domain, and further adapts the classification function to the target domain through some unlabeled target data. The theoretical analysis demonstrates its improved performance against transfer and semi-supervised algorithm. And the experiments confirm its better transferability especially for little target training data. Extending our algorithm to multiple source domains are left to the future work.

Acknowledgments. This work is supported by the National Fundamental Research Program of China (61272375, 61173100 and 61173101).

References

1. Yin, X., Han, J., Yang, J., et al.: Efficient classification across multiple database relations: A crossmine approach. *IEEE Transactions on Knowledge and Data Engineering* 18(6), 770–783 (2006)
2. Kuncheva, L.I., Rodriguez, J.J.: Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering* 19(4), 500–508 (2007)
3. Baralis, E., Chiusano, S., Garza, P.: A lazy approach to associative classification. *IEEE Transactions on Knowledge and Data Engineering* 20(2), 156–171 (2008)
4. Fung, G.P.C., Yu, J.X., Lu, H., et al.: Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering* 18(1), 6–20 (2006)
5. Al-Mubaid, H., Umair, S.A.: A new text categorization technique using distributional clustering and learning logic. *IEEE Transactions on Knowledge and Data Engineering* 18(9), 1156–1165 (2006)
6. Sarinapakorn, K., Kubat, M.: Combining subclassifiers in text categorization: A dst-based solution and a case study. *IEEE Transactions on Knowledge and Data Engineering* 19(12), 1638–1651 (2007)
7. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10), 1345–1359 (2010)

8. Arnold, A., Nallapati, R., Cohen, W.W.: A comparative study of methods for transductive transfer learning. In: Seventh IEEE International Conference on ICDM Workshops 2007, pp. 77–82 (2007)
9. Thrun, S., Mitchell, T.M.: Learning one more thing. R. Carnegie-Mellon Univ. Pittsburgh Pa Dept. of Computer Science (1994)
10. Schmidhuber, J.: On learning how to learn learning strategies (1995)
11. Caruana, R.: Multitask learning. Springer, US (1998)
12. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 120–128. Association for Computational Linguistics (2006)
13. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26(1), 101–126 (2006)
14. Pan, S.J., Zheng, V.W., Yang, Q., et al.: Transfer learning for wifi-based indoor localization. In: Association for the Advancement of Artificial Intelligence (AAAI) Workshop (2008)
15. Rosenstein, M.T., Marx, Z., Kaelbling, L.P., et al.: To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning, p. 898 (2005)
16. Dai, W., Yang, Q., Xue, G.R., et al.: Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200. ACM (2007)
17. Eaton, E., des Jardins, M.: Selective transfer between learning tasks using task-based boosting. In: Twenty-Fifth AAAI Conference on Artificial Intelligence (2011)
18. Luo, C., Ji, Y., Dai, X., et al.: Active learning with transfer learning. In: Proceedings of ACL 2012 Student Research Workshop, pp. 13–18. Association for Computational Linguistics (2012)
19. Shao, M., Castillo, C., Gu, Z., et al.: Low-Rank Transfer Subspace Learning. In: Twelfth IEEE International Conference on ICDM Workshops 2012, pp. 1104–1109 (2012)
20. Negahban, S.N., Rubinstein, B.I.P., Gemell, J.G.: Scaling multiple-source entity resolution using statistically efficient transfer learning. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 2224–2228. ACM (2012)
21. Ju, S., Li, S., Su, Y., et al.: Dual word and document seed selection for semi-supervised sentiment classification. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 2295–2298. ACM (2012)
22. Joachims, T.: Transductive inference for text classification using support vector machines. In: Machine Learning-International Workshop Then Conference, pp. 200–209. Morgan Kaufmann Publishers, Inc. (2009)
23. Chen, Y., Wang, G., Dong, S.: Learning with progressive transductive support vector machine. *Pattern Recognition Letters* 24(12), 1845–1855 (2003)
24. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
25. Joachims, T.: Learning to classify text using support vector machines: Methods, theory and algorithms. Kluwer Academic Publishers (2002)
26. Daumé III, H., Kumar, A., Saha, A.: Frustratingly easy semi-supervised domain adaptation. In: Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing, pp. 53–59. Association for Computational Linguistics (2010)

Online Distributed Passive-Aggressive Algorithm for Structured Learning

Jiayi Zhao, Xipeng Qiu*, Zhao Liu, and Xuanjing Huang

School of Computer Science, Fudan University, China
xpqiu@fudan.edu.cn

Abstract. The training phase is time-consuming for structured learning, especially for supper-tagging tasks. In this paper, we propose an online distributed Passive-Aggression (PA) by averaging parameters for parallel training, which can reduce the training time significantly. We also give theoretic analysis for its convergence. Experimental results show that our method can accelerate the training process significantly with comparable or even better accuracy.

1 Introduction

Structured learning [1] becomes a popular research topic recently. In the situation of structured learning, the labels of the data are not independent from each other, instead they form a mutually associated structure.

A cumbersome problem of structured learning is that the training time is too long, since structured learning algorithms are usually polynomial complexity in the number of labels. The larger the label set is, the more complex the algorithm will be. Traditional methods reduce the search space of the structural labels by pruning to improve computation efficiency [2]. However, these methods have a more or less performance loss.

In this paper, we propose an online distributed Passive-Aggression (PA) algorithm. We construct a parallel version of standard PA algorithm via weights averaging strategy. We also give a theoretic proof of the convergence of the training process and show that the distributed algorithm gives the same cumulative loss upper bound as the standard PA algorithm.

Our experiments show that the parallel framework gets the comparable or even better accuracy than the standard PA algorithm [3] with less training time. If the standard PA algorithm suffers from a scalability problem in both memory space and computational time when the size of a dataset is too large, our distributed PA algorithm will overcome these difficulties with less space and time using a parallel strategy and just a few machine nodes.

The rest of the paper is organized as follows. We begin by briefly reviewing the necessary background in the Passive-Aggressive algorithm in Section 2. Then we describe in detail our implementation of the distributed PA algorithm in Section 3 and give a theoretic analysis in Section 4. Finally, we report our results and analysis of experiments.

* Corresponding author.

2 Online Passive-Aggressive Algorithm

The online algorithms do not define an object function on the entire sample set because they need not obtain all samples at once. They update the parameters only depending on the observing sample. Perceptron algorithm [4] is a famous online algorithm, which is a simple but efficient and is used extensively in structured learning [5]. In contrast to perceptron, it is guided by the margin maximum idea.

Online Passive-Aggressive algorithm is a margin based online learning algorithm for various prediction tasks [3]. The difference between this algorithm and perceptron is that PA uses the margin of the samples to update the current classifier. Online PA algorithm updates the weights of features by solving a constrained optimization problem. It requires that the updated weights must stay as close as possible to the previous weights and on the other hand the updated weights correctly classify the current example with a sufficiently high margin. [6] applied this algorithm to the dependency parsing task which is a typical structured learning problem.

Given a series of samples $(\mathbf{x}_t, \mathbf{y}_t)$ denoted as \mathcal{T} , in each sample every \mathbf{x}_t has a corresponding label $\mathbf{y}_t \in \mathcal{Y}$, \mathcal{Y} is the set of all labels any \mathbf{x}_t can have. Define $\Phi(\mathbf{x}, \mathbf{y}) \in \{0, 1\}^d$ as a feature vector of d dimensions on the sample (\mathbf{x}, \mathbf{y}) . The value in each dimension of the $\Phi(\mathbf{x}, \mathbf{y})$ is a 0 – 1 value which indicates whether this feature occurs in the current sample. $\mathbf{w} \in \mathbb{R}^d$ is the parameter vector of the classifier and each dimension of it is the weight of one feature. In order to learn a proper weight vector \mathbf{w} online PA algorithm adopts an iterative method until the convergence of \mathbf{w} or the number of loops exceeds the predefined maximum iteration number.

On round t , at first the classifier predicts the labels $\hat{\mathbf{y}}_t$ for a sample \mathbf{x}_t :

$$\hat{\mathbf{y}}_t = \arg \max_{\mathbf{z}} (\mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathbf{z})), \tag{1}$$

After the prediction, the algorithm receives the correct set of relevant labels \mathbf{Y}_t and we define the margin is:

$$\gamma(\mathbf{w}_t; (\mathbf{x}_t, \mathbf{Y}_t)) = \min_{r \in \mathbf{Y}_t} \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, r) - \max_{s \notin \mathbf{Y}_t} \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, s), \tag{2}$$

From above the margin is positive if and only if all of the relevant labels are ranked higher than all of the irrelevant labels. However online PA algorithm is not satisfied by a mere positive margin as it requires the margin of every prediction to be at least a loss function $L(\mathbf{y}_t, \hat{\mathbf{y}}_t)$. The hinge-loss function which represents the loss made when the classifier gives such a prediction is defined as:

$$\ell(\mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)) = \begin{cases} 0, & \gamma(\mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)) > L(\mathbf{y}_t, \hat{\mathbf{y}}_t) \\ L(\mathbf{y}_t, \hat{\mathbf{y}}_t) - \gamma(\mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)), & \text{otherwise} \end{cases} \tag{3}$$

On round t online PA algorithm sets the new weight vector \mathbf{w}_{t+1} to be the solution to the following constrained optimization problem.

$$\begin{aligned} \mathbf{w}_{t+1} = \arg \min_{\mathbf{w} \in \mathbb{R}^d} & \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 + \mathcal{C} \cdot \xi, \\ \text{s.t. } & \ell(\mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t)) \leq \xi \text{ and } \xi \geq 0 \end{aligned} \tag{4}$$

where C is a positive parameter which controls the influence of the slack term on the objective function.

Satisfying the single constraint in the optimization problem above is equivalent to satisfying the following set of linear constraints.

$$\begin{aligned} \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, \mathbf{y}_t) - \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, \mathbf{y}) &\geq L(\mathbf{y}_t, \mathbf{y}) - \xi \\ \forall \mathbf{y} \in \{\mathcal{Y} \setminus y_t\} \end{aligned} \quad (5)$$

then the large enough margin between the correct labels and the other labels is guaranteed. Solving this constraint problem leads to a parameters update formula,

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \tau_t (\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}}_t)). \quad (6)$$

here,

$$\tau_t = \min \left\{ C, \frac{\ell(\mathbf{w}; (\mathbf{x}_t, \mathbf{y}_t))}{\|\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}}_t)\|^2} \right\} \quad (7)$$

Here C is a positive parameter which controls the influence of the slack term on the objective function. In order to prevent noise samples to make the τ_t too large to depart from the classification face, so τ_t is forced to be smaller than C .

input : training data set: $(\mathbf{x}_n, \mathbf{y}_n), n = 1, \dots, N$, and parameters: C, K
output: \mathbf{w}

Initialize: $\mathbf{c}\mathbf{w} \leftarrow 0, \mathbf{w}_0 \leftarrow 0$;

for $k = 0 \dots K - 1$ **do**

for $t = 0 \dots T - 1$ **do**

receive an example $(\mathbf{x}_t, \mathbf{y}_t)$;

predict: $\hat{\mathbf{y}}_t = \arg \max_{\mathbf{z} \neq \mathbf{y}_t} \langle \mathbf{w}_t, \Phi(\mathbf{x}_t, \mathbf{z}) \rangle$;

calculate $\ell(\mathbf{w}; (\mathbf{x}, \mathbf{y}))$;

update \mathbf{w}_{t+1} with Eq.(6);

end

$\mathbf{c}\mathbf{w} = \mathbf{c}\mathbf{w} + \mathbf{w}_T$;

end

return \mathbf{w} ;

Algorithm 1. Online Passive-Aggressive Algorithm

We assume that there exists some $\mathbf{u} \in \mathbb{R}^d$ such that $y_t(\mathbf{u} \cdot \mathbf{x}_t) > 0$ for all $t \in 1 \dots T$. In the other word \mathcal{T} are absolutely separable because such \mathbf{u} exists. So from Eq. 3 for all samples \mathbf{x}_t we can get $\ell(\mathbf{u}, \mathbf{x}_t) = 0$. Assume that for all feature function $\Phi, (\mathbf{x}, \mathbf{y}) \|\Phi(\mathbf{x}, \mathbf{y})\|^2 \leq R/2$ is satisfied. Then [3] proves the cumulative squared loss of PA algorithm on this sequence of examples is bounded by

$$\sum_{t=1}^T \ell_t^2 \leq R^2 \|\mathbf{u}\|^2 \quad (8)$$

3 Distributed Implementation of Online Passive-Aggressive Algorithm

Since the online algorithms update the parameters only depending on the observing sample, there is a straight-forward way for distributed training.

Firstly we divide the training set \mathcal{T} into S disjoint pieces $\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_S\}$ randomly. After dividing samples, all pieces are evenly assigned to different cores¹. Each core trains the parameters with PA algorithm on its assigned samples. After all cores finish training, a summarization system averages the S parameter vectors to get a new parameter vector and send it to each core for the next training iteration. The above process is repeated until the parameters converge.

This distributed online algorithm can be easily transplanted to the popular MapReduce [7] framework. The Mapper process trains parameters with each piece of samples using the Passive-Aggressive algorithm and transfer these parameters to the Reducer. The Reducer is responsible for gathering these parameters, averaging them and finally sending them to each Mapper.

Algorithm 2 is the pseudo code of the distributed Passive-Aggressive algorithm. In Algorithm 2, $\mu_{i,n}$ is a distribution which averages the parameters vector returned by each classifier. $\mathbf{w}^{(avg,n-1)}$ is the initial parameters vector on round n .

3.1 Parameters Averaging Strategy

An unsolved problem is how to mix the parameters in each iteration. In other words, we need assign a proper value of $\mu_{i,n}$ for each piece i .

Here, we discuss two different parameters averaging strategies.

One is the uniform averaging strategy, called “uniform mixing”.

Suppose that the sample set is randomly split into S portions, we set $\mu_{i,n}$ to a uniform distribution,

$$\mu_{i,n} = \frac{1}{S}. \quad (9)$$

Thus, the newer $\mathbf{w}^{(avg,n)}$ is defined as

$$\mathbf{w}^{(avg,n)} = \frac{\sum_{i=1}^S \mathbf{w}^{(i,n)}}{S}. \quad (10)$$

Another is the weighted averaging strategy, called “error mixing”. In “uniform mixing” strategy, if a machine or core is arranged to deal with a more difficultly-treated sample portion than others, the parameters obtained from it should be more important. The updating speed will be dragged down due to uniform averaging strategy. Therefore, we wish the touchy parts to contribute more in parameters updating. In the other word, we want to increase the influence of the parameters trained from those pieces with more errors. So we use the following formula to average the parameters from all portions.

$$\mathbf{w}^{(avg,n)} = \sum_{i=1}^S \frac{\delta^{i,n}}{\sum_{k=1}^S \delta^{k,n}} \mathbf{w}^{(i,n)}, \quad (11)$$

where $\delta^{i,n}$ is defined as the number of wrong classified samples in the i th training portion on the n th round.

¹ The cores can be located in different computing nodes.


```

// The Client function
// Here  $1 \leq t \leq |\mathcal{T}_i|$ 
Client( $\mathcal{T}_i : \{(\mathbf{x}_t, \mathbf{y}_t)\}$ ,  $\mathbf{w}^{(avg, n-1)}$ )
begin
  Initialize:  $\mathbf{w}^{(0)} = \mathbf{w}^{(avg, n-1)}$ ,  $k = 0$ ;
  for  $t = 1 \dots |\mathcal{T}_i|$  do
     $\hat{\mathbf{y}}_t = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} (\mathbf{w} \cdot \Phi(\mathbf{x}_t, \mathbf{y}))$ ;
    if  $\hat{\mathbf{y}}_t \neq \mathbf{y}_t$  then
       $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \tau_t (\Phi(\mathbf{x}_t, \mathbf{y}_t) - \Phi(\mathbf{x}_t, \hat{\mathbf{y}}_t))$ ;
       $k = k + 1$ ;
    end
  end
   $\mathbf{w}^{(i, n)} = \mathbf{w}^{(k)}$ ;
  output:  $\mathbf{w}^{(i, n)}$ 
end

// The Server function
// Here  $\sum_{i=1} \mu_{i, n} = 1$ 
Server( $\{\mathbf{w}^{(i, n)}, 1 \leq i \leq |\mathcal{S}|\}$ )
begin
   $\mathbf{w}^{(avg, n)} = \sum_{i=1} \mu_{i, n} \mathbf{w}^{(i, n)}$ ;
  output:  $\mathbf{w}^{(avg, n)}$ 
end

```

Algorithm 2. Distributed Passive-Aggressive Algorithm

4 Theoretical Analysis

We also need to estimate the cumulative loss produced by the distributed Passive-Aggression algorithm.

For every feature function $\Phi(\mathbf{x}, \mathbf{y})$ assume that $\|\Phi(\mathbf{x}, \mathbf{y})\|^2 \leq R/2$ is satisfied and introduce a notation $\Delta\Phi = \Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}})$.

The loss function is $\ell(\mathbf{w}) = L(\mathbf{y}, \hat{\mathbf{y}}) - (\mathbf{w} \cdot \Delta\Phi)$ following the sections above. If the final classification hyperplane produced by the classifier is $\mathbf{u} \in \mathbb{R}^d$ then we denote the result loss function as $\ell^* = \ell(\mathbf{u}) = L(\mathbf{y}, \hat{\mathbf{y}}) - (\mathbf{u} \cdot \Delta\Phi)$. We define Δ_t as the difference between the square of the distance \mathbf{w}_t apart from \mathbf{u} and the square of the distance \mathbf{w}_{t+1} apart from \mathbf{u} on round t ,

$$\begin{aligned}
\Delta_t &= \|\mathbf{w}_t - \mathbf{u}\|^2 - \|\mathbf{w}_{t+1} - \mathbf{u}\|^2 \\
&= \|\mathbf{w}_t - \mathbf{u}\|^2 - \|\mathbf{w}_t + \tau_t \Delta\Phi - \mathbf{u}\|^2 \\
&= -2\tau_t (\mathbf{w}_t - \mathbf{u}) \cdot \Delta\Phi - \tau_t^2 \|\Delta\Phi\|^2 \\
&\geq 2\tau_t (\ell_t - L(\mathbf{y}, \hat{\mathbf{y}}) - (\ell_t^* - L(\mathbf{y}, \hat{\mathbf{y}}))) - \tau_t^2 \|\Delta\Phi\|^2 \\
&= \tau_t (2\ell_t - \tau_t \|\Delta\Phi\|^2 - 2\ell_t^*)
\end{aligned} \tag{12}$$

If the sample set $\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_t, \mathbf{y}_t)\}$ is linearly separable, we will get $\ell^* = 0$ is satisfied for any sample. Then

$$\Delta_t \geq \tau_t(2l_t - \tau_t \|\Delta\Phi\|^2) = l_t^2 / \|\Delta\Phi\|^2 \geq l_t^2 / R^2 \quad (13)$$

from above,

$$\ell_t^2 / R^2 \leq \|\mathbf{w}_t - \mathbf{u}\|^2 - \|\mathbf{w}_{t+1} - \mathbf{u}\|^2 \quad (14)$$

So the upper bound of the loss of a sample is determined by the difference of the parameters and the final classification face \mathbf{u} before and after each round.

Now consider the cumulative loss after weighted average on round n ,

$$\sum_{i=1}^S \mu_{i,n} \sum_{t_i=1}^{T_i} \frac{l_{t_i,n}^2}{R^2} \leq \|\mathbf{w}^{(avg,n-1)} - \mathbf{u}\|^2 - \sum_{i=1}^S \mu_{i,n} \|\mathbf{w}^{(i,n)} - \mathbf{u}\|^2 \quad (15)$$

According to the algorithm 2 we know $\mathbf{w}^{(avg,n)} = \sum_{i=1}^S \mu_{i,n} \mathbf{w}^{(i,n)}$ and use Jensen's inequality. Δ^2 is denoted as $\|\mathbf{w}^{(avg,n)} - \mathbf{u}\|^2$,

$$\begin{aligned} \Delta^2 &= \left\| \sum_{i=1}^S \mu_{i,n} \mathbf{w}^{(i,n)} - \mathbf{u} \right\|^2 \\ &\leq \sum_{i=1}^S \mu_{i,n} \|\mathbf{w}^{(i,n)}\|^2 - 2 \sum_{i=1}^S \mu_{i,n} \mathbf{u} \cdot \mathbf{w}^{(i,n)} + \|\mathbf{u}\|^2 \\ &= \sum_{i=1}^S \mu_{i,n} \|\mathbf{w}^{(i,n)} - \mathbf{u}\|^2 \end{aligned} \quad (16)$$

then

$$\|\mathbf{w}^{(avg,n)} - \mathbf{u}\|^2 \leq \sum_{i=1}^S \mu_{i,n} \|\mathbf{w}^{(i,n)} - \mathbf{u}\|^2 \quad (17)$$

So the cumulative loss in N iterations is

$$\begin{aligned} &\sum_{n=1}^N \sum_{i=1}^S \mu_{i,n} \sum_{t_i=1}^{T_i} \frac{l_{t_i,n}^2}{R^2} \\ &\leq \|\mathbf{w}^{(avg,0)} - \mathbf{u}\|^2 - \sum_{i=1}^S \mu_{i,N} \|\mathbf{w}^{(i,N)} - \mathbf{u}\|^2 \\ &\leq \|\mathbf{w}^{(avg,0)} - \mathbf{u}\|^2 \\ &= \|\mathbf{u}\|^2 \end{aligned} \quad (18)$$

so

$$\sum_{n=1}^N \sum_{i=1}^S \mu_{i,n} \sum_{t_i=1}^{T_i} l_{t_i,n}^2 \leq R^2 \|\mathbf{u}\|^2 \quad (19)$$

Refer to the section 2 we can find that the online distributed Passive-aggression algorithm has the same cumulative loss upper bound.

5 Experiments

We verify the efficiency of our method with joint segmentation and POS-tagging problem, which is a common task in natural language processing (NLP). Structured learning methods are usually adopted to solve this problem nowadays but suffers from long training time.

The construction of our distributed algorithm is based on the FudanNLP toolkit [8] for natural language processing.

We use POS tagging task with CTB corpus from the SIGHAN Bakeoff [9]. There are 37 kinds of POS tags in this corpus. We use cross-label method for this task. The cross-label consists of two parts: the first part is the label of segmentation, and the second part is the label of POS tag. We use four segmentation labels (“B”, “M”, “E”, “S”) to represent the beginning, middle, end of a word and a single character word respectively. For example, a label “B-NN” means the beginning of a noun word. After processing the corpus, we get 108 different cross-labels.

Table 1 shows all the feature templates used in our task. These templates are very common for the segmentation and POS-tagging task.

Table 1. Feature Templates

(1.1)	$c_{i-2}t_i, c_{i-1}t_i, c_it_i, c_{i+1}t_i, c_{i+2}t_i$
(1.2)	$c_{i-1}c_it_i, c_ic_{i+1}t_i, c_{i-1}c_{i+1}t_i$
(1.3)	$t_{i-1}t_i$

Note: c_i represents the character at position i in the sentence and t_i is for the target label of this character.

In our experiments, 23, 443 sentences are used for training and 2, 079 sentences are used for testing. We split the training set into 2, 5, 10, 20 and 50 pieces randomly and train each piece in each machine or core.

We firstly compare the different averaging strategy of parameters.

Figure 1 gives the comparison between the uniform mixing and error mixing methods. In this experiment we split the entire sample set into 2 pieces and 5 pieces respectively and inspect the training errors at each iteration.

Figure 1 shows that on each round the numbers of training errors are almost the same for both 2 pieces and 5 pieces training strategy. This is because training corpus is randomly and uniformly split so the probability of hard tagging sentences gathering in a single piece is rare. So the number of training errors of each machine is very close. The difference between the two averaging strategies is very small in this situation. However a better averaging strategy maybe speeds the process of training up.

In the later evaluation, we just use the “uniform mixing” strategy.

Table 2 gives the results of our method and the standard PA algorithm with different experimental settings. It shows that our algorithm can increase the speed of the training without accuracy loss.

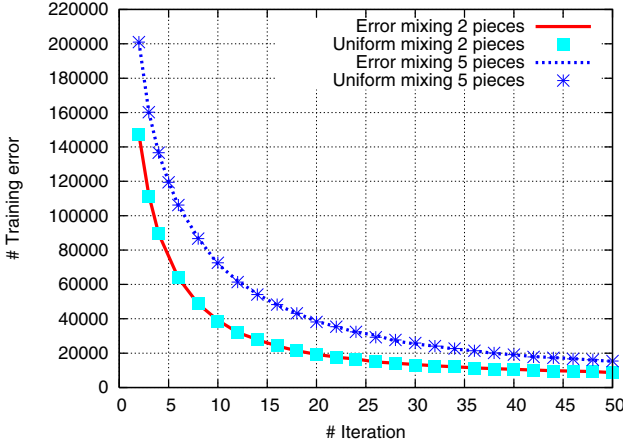


Fig. 1. Training errors with iterations. Uniform mixing just uniformly averages the weights from each machine and Error mixing assigns $\mathbf{w}^{(avg,n)} = \sum_{i=1}^S \frac{\delta^{i,n-1}}{\sum_{k=1}^S \delta^{k,n-1}} \mathbf{w}^{(i,n-1)}$ as the new weight using weighted average

Table 2. Evaluation of trained models

Methods	Segmentation		POS-tagging		Time(s) [†]
	Accuracy(%)	F1	Accuracy(%)	F1	
original	97.63	95.00	89.67	87.54	6679.3
2 pieces	97.54	94.95	89.39	87.36	3514.8
2 pieces*	97.53	94.90	89.18	87.01	3414.2
5 pieces	97.76	95.35	89.93	87.78	1839.9
5 pieces*	97.78	95.37	90.04	87.88	1259.4
10 pieces	97.82	95.45	90.05	87.91	761.6
20 pieces	97.81	95.44	90.12	88.10	959.9 [‡]
50 pieces	97.85	95.52	90.32	88.34	654.5

* The method with * uses error mixing averaging strategy.

† This is training time omitting the time cost by data transferring.

‡ When we split the sample set into 20 pieces the algorithm didn't convergence after running for 150 iterations.

The entries in the Table 2 with star ‘*’ is the ‘‘Error Mixing’’ version of distributed online PA algorithm. As the result from Table 2 the distributed algorithm get approximate or even better result of the standard PA algorithm.

We also compared the performance of the training process and the convergence property of the normal online PA algorithm with our distributed implementation. Figure 2 displays relations between the training precision and the number of iterations under different pieces of samples. We set the maximum iterations to be 150. We can see that it needs the more iterations to convergence as the number of pieces increases. However,

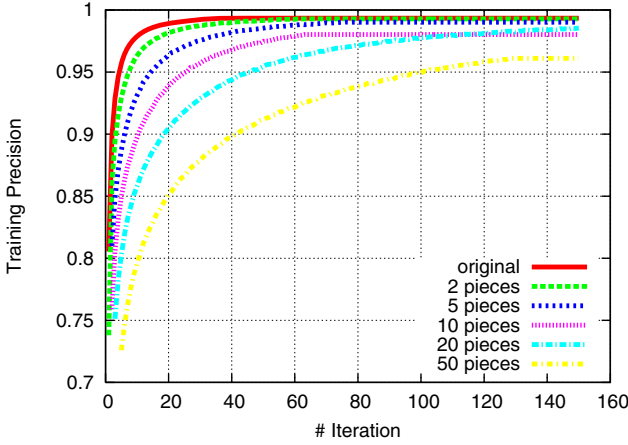


Fig. 2. Training precision with iterations

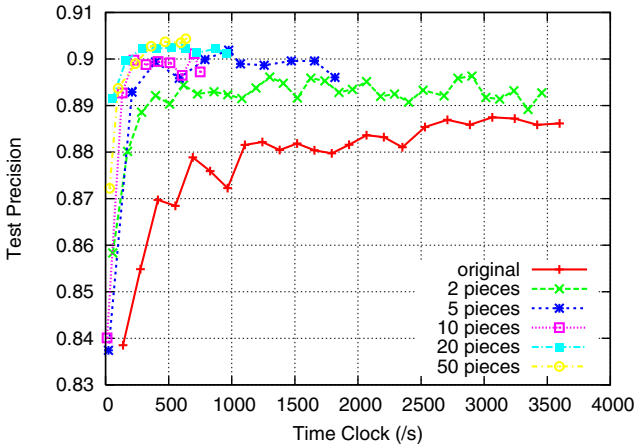


Fig. 3. Testing precision with training time

since each piece use less time for the larger number of pieces, we can reduce the training time in total.

To evaluate the effect of speed-up in training phrase, we compare the test precision as the time goes on. In this section we ignore the time of network I/O operations and only consider the accumulated CPU time. The comparisons are shown in Figure 3. Apparently, the distribution PA algorithm is faster than the original algorithm to achieve the same test precision. The parallel PA algorithm is also faster to converge with comparable precision to the normal PA algorithm. We can see that the test precision of distributed PA algorithm is a little higher.

5.1 Experimental Discussion

There are some things to be worth noting from our experiments.

Firstly, the distributed online PA algorithm has a little higher testing accuracy than the standard one mentioned above. We suspect this happens for the reason that the distributed method is a form of parameter averaging which has the same effect as the average perceptron [5]. Although training time is dramatically increased after we add an average strategy to the original algorithm, the test accuracy is still lower than the distributed algorithm. Maybe this happens because the original version overfits the training data.

Secondly, when the training set is split into large enough pieces, the time in the testing data is almost the same to converge and to achieve the best performance. This is shown in Figure 3 that lines in the left up corner cover each other. Further considering the network I/O waste, an unlimited split of the sample set is undesirable. The segmentation with 10 pieces is a better choice for POS tagging in our paper.

6 Related Works

The distributed machine learning is attracting increasing attention in recent years. For example, Mahout² is a famous package of scalable parallel machine learning libraries based on MapReduce [7] framework. The distributed computing framework also provides a new way to deal with the large computation cost in structured learning.

Chu et al.[10] develop a broadly applicable parallel programming method, which is easily applied to many different learning algorithms. They have also investigated parallel implements of many batch algorithms and modified these algorithms in the map-reduce framework. For online algorithms, parameters updating process is a serial procedure. In order to guarantee the accuracy and convergence of online algorithms we need a particular strategy to make the serial process parallel.

Chang et al.[11] develop a parallel support vector machine (SVM)[12] algorithm which reduces memory use through performing a row-based, approximate matrix factorization.

Wolfe et al.[13] present a framework which fully distributes the EM procedure. In their implementation every node only interacts with parameters relevant to its data and sends messages to other nodes along a junction-tree topology.

Chiang et al.[14] parallel the MIRA[15] online training algorithm with some coordination mechanism among processors. However, there is no detailed theoretic discussion in their paper.

McDonald et al.[16] propose a parameters mixing strategy used in the parallel Perceptron implementation. Their work is very similar to ours, but PA and Perceptron are different online algorithms. So the theoretic proofs of the convergence are actually quite different.

7 Conclusion

In this paper we propose a distributed training strategy with the online PA algorithm for structured learning. We guarantee its convergence with the proposed averaging strategy.

² <http://mahout.apache.org/>

Experimental results show that our method significantly reduces the time cost without decreasing the accuracy. Although the number of iterations increases, less time cost in each iteration also make the training time still less.

In the future, we wish to extend our algorithm to more complex structured learning, such as parsing.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments. This work was funded by NSFC (No.61003091 and No.61073069).

References

1. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.* 6, 1817–1853 (2005)
2. Zhang, Y., Clark, S.: A fast decoder for joint word segmentation and POS-tagging using a single discriminative model. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010*, pp. 843–852. Association for Computational Linguistics, Stroudsburg (2010)
3. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7, 551–585 (2006)
4. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65(6), 386–408 (1958)
5. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (2002)
6. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: *Proc. of HLT-EMNLP* (2005)
7. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. *Commun. ACM* 51, 107–113 (2008)
8. Qiu, X., Zhang, Q., Huang, X.: FudanNLP: A toolkit for Chinese natural language processing. In: *Proceedings of ACL* (2013)
9. Jin, C., Chen, X.: The fourth international Chinese language processing bakeoff: Chinese word segmentation, named entity recognition and Chinese pos tagging. In: *Sixth SIGHAN Workshop on Chinese Language Processing*, p. 69 (2008)
10. Chu, C.T., Kim, S.K., Lin, Y.A., Ng, A.Y.: Map-reduce for machine learning on multicore. *Architecture* 19, 281 (2007)
11. Chang, E.Y., Zhu, K., Wang, H., Bai, H., Li, J., Qiu, Z., Cui, H.: Psvm: Parallelizing support vector machines on distributed computers. *Change* 20(2), 1–8 (2007)
12. Cristianini, N., Shawe-Taylor, J.: *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge Univ. Pr. (2000)
13. Wolfe, J., Haghighi, A., Klein, D.: Fully distributed em for very large datasets. In: *Proceedings of the 25th International Conference on Machine Learning, ICML 2008*, pp. 1184–1191. ACM, New York (2008)
14. Chiang, D., Marton, Y., Resnik, P.: Online large-margin training of syntactic and structural translation features. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 224–233. Association for Computational Linguistics (2008)
15. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research* 3, 951–991 (2003)
16. McDonald, R., Hall, K., Mann, G.: Distributed training strategies for the structured perceptron. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 2010*, pp. 456–464. Association for Computational Linguistics, Stroudsburg (2010)

Power Law for Text Categorization

Wuying Liu¹, Lin Wang², and Mianzhu Yi¹

¹ PLA University of Foreign Languages, 471003 Luoyang, Henan, China
wyliu@nudt.edu.cn, mianzhuyi@gmail.com

² National University of Defense Technology, 410073 Changsha, Hunan, China
wanglin@nudt.edu.cn

Abstract. Text categorization (TC) is a challenging issue, and the corresponding algorithms can be used in many applications. This paper addresses the online multi-category TC problem abstracted from the applications of online binary TC and batch multi-category TC. Most applications are concerned about the space-time performance of TC algorithms. Through the investigation of the token frequency distribution in an email collection and a Chinese web document collection, this paper re-examines the power law and proposes a random sampling ensemble Bayesian (RSEB) TC algorithm. Supported by a token level memory to store labeled documents, the RSEB algorithm uses a text retrieval approach to solve text categorization problems. The experimental results show that the RSEB algorithm can achieve the state-of-the-art performance at greatly reduced space-time requirements both in the TREC email spam filtering task and the Chinese web document classifying task.

Keywords: Text Categorization, Power Law, Online Binary TC, Batch Multi-Category TC, TREC.

1 Introduction

Automated text categorization (TC) has been widely investigated since the early days of artificial intelligence. According to the arriving mode of documents, TC can be divided into online TC and batch TC. According to the number of predefined categories, TC includes binary TC and multi-category TC. For instance, email spam filtering is an online binary TC application and web document classifying is normally a batch multi-category TC application. The binary TC is a special case of the multi-category TC and a batch TC can be regarded as a series of online classifications, so this paper addresses the online multi-category TC problem.

Most TC applications pay more attention to the space-time complexity of TC algorithms. The power law of word frequency in a set of text documents, a famous random distribution phenomenon, was discovered for a long time. How to use the power law to reduce the space-time complexity of statistical TC algorithms is a significant research problem. In statistical TC algorithms, token frequency is a very effective feature. If we only use token frequency features in a closed text collections, the feature with once occurrence will never be used, and according to the power law,

we can easily remove these useless long tail features for lower space-time costs. But in an online situation, we meet a puzzle of open feature space. The ubiquitous power law may bring an opportunity to propose a novel statistical TC algorithm for the efficient online multi-category TC problem.

The rest of this paper is organized as follows. In section 2, we describe some related works about TC. In section 3, we investigate the power law of token frequency both in an email collection and a web document collection, and analyze the potential uselessness rate. In section 4, we propose a random sampling ensemble Bayesian (RSEB) algorithm. In section 5, the experiment and result are described. At last, in section 6, the conclusion and further work are given.

2 Related Work

Recently, statistical TC algorithms have been widely used in TC applications [1]. Email spam filtering is defined as an online supervised binary TC problem, which is simulated as an immediate full feedback task (IFFT) in the TREC spam track. Web document classifying is normally defined as a batch multi-category TC problem, which is simulated as a 12-category Chinese web document classifying task (WDCT) [2].

Many online binary TC algorithms have been proposed for the email spam filtering. For instance: 1) based on the vector space model (VSM), the online Bayesian algorithm uses the joint probabilities of words and categories to estimate the probabilities of categories for a given document; 2) the relaxed online support vector machines (SVMs) algorithm [3] relaxes the maximum margin requirement and produces nearly equivalent results, which has gained several best results in the TREC 2007 spam track; and 3) the online fusion of dynamic Markov compression (DMC) and logistic regression on character 4-gram algorithm [4] is the winner on the IFFT in the TREC 2007 spam track.

Many batch multi-category TC algorithms have been introduced to deal with the web document classifying. For instance: 1) the k-nearest neighbor (kNN) TC algorithm decides a document according to the k nearest neighbor document categories; 2) the centroid TC algorithm [5] is based on the assumption that a given document should be assigned a particular category if the similarity of this document to the centroid of its true category is the largest; and 3) the winnow algorithm [6] uses a multiplicative weight-update scheme that allows it to perform much better when many dimensions are irrelevant.

Structured feature and token frequency distribution feature of documents are both crucial to the classification performance. Previous research shows that the multi-field structured feature of email documents supports the divide-and-conquer strategy, and can be used to improve the classification performance [7]. This multi-field learning (MFL) framework will bring the statistical, computational and representational advantages like that of ensemble learning methods [8]. Previous research also shows that the token frequency distribution follows the power law [9], which is a prevalent random phenomenon in many text documents.

The previous TC algorithms often pursue the high classification accuracy and the high overall performance of supervised learning, without more claiming their low space-time complexity. However, in practice the algorithm is space-time-cost-sensitive for many real-world large-scale applications. For instance, specified in the TREC spam track, the space-time limitation (total 1 GB RAM and 2 sec/email) is still

unpractical and horrible in a real large-scale email system, where large-scale emails will form a round-the-clock data stream and there will be more than thousands of emails arriving during 2 seconds. Especially, it is unreasonable to require an industrial TC algorithm with a time-consuming training or updating: such a requirement defeats previous complex statistical algorithms, and motivates us to explore a space-time-efficient TC algorithm.

3 Re-examination of Power Law

3.1 Corpora

The email documents corpus is the TREC07p collection, firstly designed as a public corpus for TREC 2007 spam track, which contains total 75,419 emails (25,220 hams and 50,199 spams). Each email document is stored as a plain text file, and email text is unaltered except for a few systematic substitutions of names.

The Chinese web documents corpus is the TanCorp collection, which contains total 14,150 documents and is organized in two hierarchies. The first hierarchy contains 12 big categories and the second hierarchy consists of 60 small classes. In this paper, we use TanCorp-12.

From the perspective of lingual category, above two corpora are representative. The TREC07p corpus contains multi-language, although the main language is English. The TanCorp corpus is a Chinese text documents collection.

3.2 Token Frequency Distribution

In order to re-examine token frequency distribution, we calculate the number of tokens. According to the widely-used VSM, a text document is normally represented as a feature vector, and each feature is a text token. Previous research has shown that overlapping word-level k-gram token model can achieve promising results [10]. But different languages have different appropriate k values, and the different representational granularities determine the total number of text features. Here, we consider four overlapping word-level k-gram token models (1-gram, 2-gram, 3-gram, 4-gram) to represent tokens.

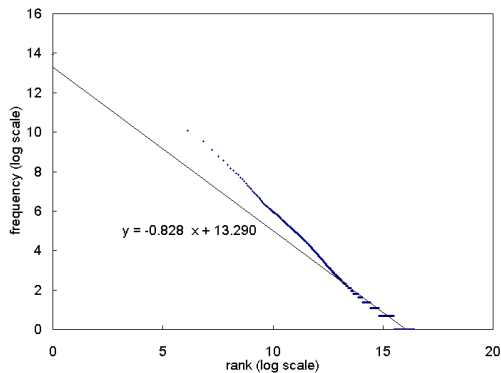


Fig. 1. Word-level 4-gram Token Frequency-Rank in the TREC07p Collection

Firstly, we regard an email message as a single plain-text document, and calculate the number of each token occurrence in the TREC07p collection. Fig. 1 shows the token frequency as the function of the token's rank with the word-level 4-gram token model. The horizontal-axis (x-axis) indicates the token's rank (log scale), and the vertical-axis (y-axis) indicates the token frequency (log scale). The trendline ($y = a x + b$) indicates that the frequency distribution of the word-level 4-gram token approximately follows a power law.

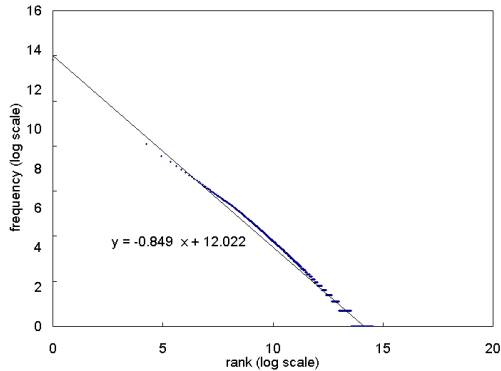


Fig. 2. Word-level 2-gram Token Frequency-Rank in the TanCorp Collection

Secondly, we use the same method to calculate the number of each token occurrence in the TanCorp collection. Fig. 2 shows the token frequency as the function of the token's rank with the word-level 2-gram token model. The trendline also shows a power law distribution.

Table 1. Trendline Coefficients in TREC07p Collection and TanCorp Collection

		1-gram	2-gram	3-gram	4-gram
TREC07p	a	-1.118	-1.103	-0.923	-0.828
	b	15.047	16.454	14.501	13.290
TanCorp	a	-1.766	-0.849	-0.460	-0.280
	b	20.129	12.022	6.879	4.242

Finally, the statistical results show that not only the 4-gram token frequency distribution in the TREC07p collection and the 2-gram token frequency distribution in the TanCorp collection follow the power law, but the others k-gram token frequency distribution also follow the power law. Table 1 shows the detailed trendline ($y = a x + b$) coefficients a and b .

Above re-examination shows that the token frequency distribution follows the power law in the multilingual email documents, the Chinese web documents, and the field sub-documents of email [11]. The ubiquitous power law indicates that the weightiness of each token feature is not equivalent, which suggests a feature selection method to remove those useless features for lower space-time costs.

3.3 Potential Useless Feature

In statistical TC algorithms, the text feature selection is a widely-used method against the high dimensional problem and has a crucial influence on the classification performance. The iteration, the cross-validation and the multi-pass scans are all effective methods to the text feature selection. But these methods bring the high space-time complexity. If we can detect and remove those useless features, we will save more time and space. However, what is the useless feature and how to find it?

In a whole text documents set, a token feature with a less frequency (≤ 2) is a potential useless feature. As an extreme instance, if a token feature occurs only once all the time, it is useless because it will never be used in the future. So the useful features will not decrease after removing the useless features. We further define the uselessness rate R_u as the ratio of the number of token features with less frequency to the total number of token features. Here, we only consider the word-level 4-gram token in the TREC07p collection and the word-level 2-gram token in the TanCorp collection.

Table 2. Feature Number and Uselessness Rate

	Feature Number (num)			$R_u(\%)$	
	$N(1)$	$N(2)$	$N(*)$	$R_u(\leq 1)$	$R_u(\leq 2)$
TREC07p	9,985,998	2,885,917	15,754,699	63	82
TanCorp	1,316,422	325,834	2,087,815	63	79

The $N(1)$ and $N(2)$ separately denote the number of token features which only occurs once and twice in the related documents set. The $N(*)$ denotes the total number of token features. The $R_u(\leq 1)$ is defined as $N(1)/N(*)$, and the $R_u(\leq 2)$ is defined as $(N(1)+N(2))/N(*)$. Table 2 shows that the uselessness rate in the TREC07p collection is between 63% and 82%, and the uselessness rate in the TanCorp collection is between 63% and 79%. The uselessness rates are all higher in the five natural text fields of the TREC07p corpus [11]. If we can get the whole text documents before TC predicting, we will easily find these useless token features and cut this long tail. However, the online TC application faces an open text space problem, and we can not foreknow a token feature's occurrence in the future. Though an online text stream makes it impossible to find these posteriori useless token features, the higher uselessness rate indicates that there are lots of useless token features. Supported by the priori and ubiquitous power law, this paper proposes a random sampling method to remove these useless token features at the time of online training. The range of uselessness rate indicates the theoretical tolerant range of training feature loss rate.

4 Random Sampling Ensemble Bayesian Algorithm

4.1 Token Level Memory

In this paper, the object categories of the online multi-category TC problem are represented as a set in the form ($C=\{C_i\}$, $i=1, 2, \dots, n$), and a document D is represented as a sequence of tokens in the form ($D=T_1T_2\dots T_j\dots$). Here, we use the overlapping word-level k-gram model to define a token. The token frequency within

historical labeled documents, the key feature of online supervised machine learning, implies rich classification information and must be stored effectively. The token level memory (TLM) is a data structure to store the token frequency information of labeled documents, from which we can conveniently calculate the Bayesian conditional probability $P(C_i|T_j)$ for the object category C_i and the token T_j . We straightforwardly combine the Bayesian conditional probabilities of tokens and choose the category of the biggest probability as the document’s final category prediction.

Fig. 3 shows the TLM structure, including two indexes organized as two hash tables. The table entry of the DF index is a key-value pair $\langle key_C, value \rangle$, where each key C_i denotes the i th category and each value $DF(C_i)$ denotes the total number of documents with C_i category labels. The hash function $hash_{DF}(C_i)$ maps the category C_i to the address of the $DF(C_i)$. The table entry of the TF index is also a key-value pair $\langle key_T, value \rangle$, where each key T_j denotes a token and each value consists of n integers. The integer $TF_i(T_j)$ denotes the occurrence times of the token T_j in labeled C_i category documents. The hash function $hash_{TF}(T_j)$ maps the token T_j to the address of the n integers.

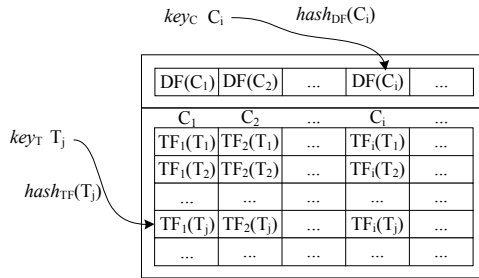


Fig. 3. Token Level Memory

The TLM stores labeled tokens, the tiny granularity labeled examples, while other memory-based algorithms, such as kNN, store document-level labeled examples. This index structure has a native compressible property of raw texts. Each incremental updating or retrieving of index has a constant time complexity. The power law can help us to remove lots of long tail tokens through random sampling learning.

4.2 Random Sampling Learning

Supported by the TLM, the RSEB algorithm takes the online supervised training process as an incremental updating process of indexes, and takes the online predicting process as a retrieving process of indexes.

According to the power law, we add a random sampling learning into the online supervised training process. The random sampling idea is based on the assumption that some tokens selected randomly according to equiprobability trend to be higher frequency tokens. If only the relative frequency features are concerned among tokens, we can use partial tokens of a labeled document to update the TLM after random sampling. As a result, lots of long tail tokens will be online removed, and the relative frequency will not change among tokens. We define the random sampling rate R_{rs} as the ratio of the number of tokens added into the TLM to the total number of tokens of each labeled document, which is a real number ($R_{rs} \in [0, 1]$).

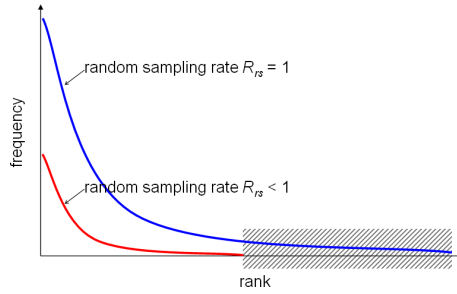


Fig. 4. Random Sampling Sketch

Fig. 4 shows the random sampling sketch. The horizontal-axis (x-axis) indicates the token's rank, and the vertical-axis (y-axis) indicates the token frequency. If $R_{rs}=1$, all the tokens of a labeled document will be added into the TLM at the time of online training. While if $R_{rs}<1$, there will be some tokens absent in the TLM. Along the online incremental updating, these above two cases will form two power law curves in Fig. 4, where the shadow range denotes removed tokens. These two power law curves also indicate that the random sampling will not change the total distribution of the relative frequency among tokens. Of course, if the random sampling rate approximates zero, the classification ability of the TLM will also be damaged. However, what is the optimal random sampling rate? Theoretically, a promising random sampling rate is the $(R_{rs}=1-R_u)$. But the exact R_u is also not a priori value. Fortunately, the ubiquitous power law gives an approximate heuristic, such as the 20/80 rule of the $R_{rs}|R_u$.

```

//OTP: Online Training Procedure
OTP(Document  $d$ ; Gram  $k$ ; Category  $c$ ; TLM  $t$ ;  $R_{rs}$   $r$ )
(1) String[]  $T := \text{Tokenizer}(d; k)$ ;
(2) String[]  $T_{rs} := \text{RandomSampler}(T; r)$ ;
(3)  $t.DF(c) := t.DF(c)+1$ ;
(4) Loop: For Each  $T_j \in T_{rs}$  Do:
  (4.1) If  $t.\text{containKey}_T(T_j)$  Then:  $t.TF(c, T_j) := t.TF(c, T_j)+1$ ;
  (4.2) Else:
    (4.2.1)  $t.TF(c, T_j) := 1$ ;
    (4.2.2)  $t.TF(\sim c, T_j) := 0$ ; //  $\sim c$  means all other categories
    (4.2.3)  $t.\text{putKey}_T(T_j)$ ;
(5) Output:  $t$ .

//Extract tokens based on overlapping word-level  $k$ -gram model
Tokenizer(Document  $d$ ; Gram  $k$ )

//Sample tokens based on the random sampling rate  $R_{rs}$ 
RandomSampler(String[]  $T$ ;  $R_{rs}$   $r$ )

```

Fig. 5. Pseudo-Code for Online Training

Fig. 5 gives the pseudo-code for the online training procedure of the RSEB algorithm. When a new labeled document arrives, the online training procedure only needs to add the document's tokens into the TLM. This procedure firstly analyzes the

document text and extracts tokens based on an overlapping word-level k -gram model, and then randomly samples the tokens based on a preset random sampling rate, and finally updates the token frequency or adds a new index entry to the TLM according to the tokens after the random sampling.

4.3 Ensemble Bayesian Predicting

The Bayesian conditional probability predicting is a very classical method. According to each observed token of a document, the Bayesian method can obtain an array of probabilities, reflecting the likelihood that the classified document belongs to each category. The ensemble method uses arithmetical average to combine the multi-array of probabilities predicting from all tokens to form a final array. And then, the category of the maximal probability in the final array is predicted as the document's category.

```

//OPP: Online Predicting Procedure
OPP(Document  $d$ ; Gram  $k$ ; TLM  $t$ )
(1) String[]  $T :=$  Tokenizer( $d$ ;  $k$ );
(2) Float[]  $ep :=$  new Float[ $n$ ];
(3) Loop: For Each  $T_j \in T$  Do:
    (3.1) Float[]  $p :=$  BayesianPredictor( $T_j$ ;  $t$ );
    (3.2) Loop: For Each  $i \in [1, n]$  Do:
        (3.2.1)  $ep[i] := ep[i] + p[i]$ ;
(4) Float  $sum :=$  Sum( $ep$ ); //Add the  $n$  floats to a  $sum$ 
(5) Loop: For Each  $i \in [1, n]$  Do:
    (5.1)  $ep[i] := ep[i] / sum$ ;
(6) Integer  $index :=$  Math.max( $ep$ ).getIndex();
(7) Output:  $C_{index}, ep[index]$ .

//Compute conditional probability  $P(C_i|T_j)$  for each category  $C_i$ 
BayesianPredictor(String  $token$ ; TLM  $t$ )
(1) Float[]  $p :=$  new Float[ $n$ ];
(2) Loop: For Each  $i \in [1, n]$  Do:
    (2.1)  $p[i] := t.TF(C_i, token) / t.DF(C_i)$ ;
(3) Float  $sum :=$  Sum( $p$ ); //Add the  $n$  floats to a  $sum$ 
(4) Loop: For Each  $i \in [1, n]$  Do:
    (4.1)  $p[i] := p[i] / sum$ ;
(5) Output:  $p$ .

//Extract tokens based on overlapping word-level  $k$ -gram model
Tokenizer(Document  $d$ ; Gram  $k$ )

```

Fig. 6. Pseudo-Code for Online Predicting

Fig. 6 gives the pseudo-code for the online predicting procedure of the RSEB algorithm. When a new document arriving, the online predicting procedure is triggered: 1) the procedure also analyzes the document text and extracts tokens based on an overlapping word-level k -gram model; 2) the procedure retrieves the current TLM and calculates each token's probabilities array according to the Bayesian conditional probability; 3) the procedure assumes that each token's contribution is equivalent to the final probabilities array and uses the arithmetical average method to

calculate a final ensemble probabilities array; and 4) the procedure chooses the maximal probability in the final ensemble probabilities array, and outputs the document's category predication and this maximal probability.

4.4 Space-Time Complexity

The RSEB algorithm mainly makes up of the online training and the online predicting procedures, whose space-time complexity depends on the TLM storage space and the loops in the two procedures.

The TLM storage space is efficient owing to two reasons: the native compressible property of index files [12] and the random-sampling-based compressible property at the time of online incremental updating. Hash list structure, prevalingly employed in information retrieval, has a lower compression ratio of raw texts. Though the training documents will mount in the wake of the increasing of online feedbacks, the TLM storage space will only increase slowly. The native compressible property of index files ensures that the TLM storage space is theoretically proportional to the total number of tokens, and not limited to the total number of training documents. The random-sampling-based compressible property of TLM is caused by the power law of token frequency distribution and the only requirement of relative frequency. The random-sampling-based feature selection can cut the long tail useless features in the online situation. The above two compressible properties make that the online labeled document stream can be incrementally space-efficiently stored.

The incremental updating or retrieving of TLM has a constant time complexity according to hash functions. The online training procedure is lazy, requiring no retraining when a new labeled document added. Fig. 5 shows that the time cost of per updating is only proportional to the total number of tokens in the document. Except the loop (see (4) of Fig. 5) according to the number of tokens, there are no time-consuming operations. The major time cost of the online predicting procedure is related to the number of categories. The straightforward calculating makes that the time complexity is acceptable in the practical online application.

5 Experiment

5.1 Implementation

We implement an email spam filter (*esf*) and a web document classifier (*wdc*) according to the proposed RSEB algorithm. In the filter and classifier, we do nothing about text pre-processing, such as stemming, stop word elimination, etc.

Using *cs4* combining strategy [7], the *esf* filter is combined from seven field classifiers within the seven-field MFL framework, five natural fields (Header, From, ToCcBcc, Subject, and Body) and two artificial fields (H.IP, H.EmailBox), and each field classifier is an implementation of the RSEB algorithm with binary categories. In each field classifier, the overlapping word-level 4-gram token model is applied.

Applying the overlapping word-level 2-gram token model, the *wdc* classifier is an implementation of the RSEB algorithm with 12 categories in Chinese texts. In order to extract word-level tokens, we build a Chinese segmenter in the *wdc* classifier.

5.2 Task and Evaluation

We run an IFFT of email spam filtering and a WDCT of 12-category Chinese web document classifying to evaluate the performance of the RSEB algorithm.

On the email spam filtering, we report the overall performance measurement 1-ROCA, the area above the receiver operating characteristic (ROC) curve percentage, where 0 is optimal, to evaluate the filter's performance. We compare the *esf* to the *bogo* filter (bogo-0.93.4), the *tftS3F* filter, and the *wat3* filter on the IFFT, defined in the TREC spam track. The *bogo* filter is a classical implementation of online Bayesian algorithm, the *tftS3F* filter is based on the relaxed online SVMs algorithm and has gained several best results in the TREC 2007 spam track, and the *wat3* filter is based on the online fusion of DMC and logistic regression algorithm, which is the winner on the IFFT in the TREC 2007 spam track. In this experiment, we use the TREC07p corpus, the TREC spam filter evaluation toolkit, and the associated evaluation methodology.

On the web document classifying, we use three-fold cross validation in the experiments by evenly splitting the TanCorp-12 dataset into three parts and use two parts for training and the remaining third for testing. We perform the training-testing procedure three times and use the average of the three performances as the final result. Here reports classical MacroF1 and MicroF1 measures. We run the *wdc* classifier on the 12-category Chinese WDCT, and compare the results of the *wdc* classifier as well as to that of the *kNN* classifier, the *centroid* classifier, and the *winnow* classifier.

The hardware environment for running experiments is a PC with 1 GB memory and 2.80 GHz Pentium D CPU.

5.3 Results and Discussions

There are four experiments. On the email spam filtering, the experiment A tries to evaluate that the RSEB algorithm is time-efficient and can achieve the best overall performance, and the experiment B wants to verify that the TLM data structure has the random-sampling-based compressible property and the proposed random sampling method is space-efficient. On the web document classifying, the experiment C tries to evaluate the effectiveness of the RSEB algorithm, and the experiment D wants to verify the random-sampling-based compressible property of the TLM data structure in the multi-category situation.

In the experiment A, the *bogo*, *tftS3F*, and the *esf* filter run on the IFFT on the TREC07p corpus separately, and the *esf* filter sets its random sampling rate $R_s=1$. The detailed experimental results are showed in Table 3. The results show that the *esf* filter can complete filtering task in high speed (2,834 sec), whose overall performance 1-ROCA is comparable to the best *wat3* filter's (0.0055) among the participators at the TREC 2007 spam filtering evaluation. The time and 1-ROCA performance of the *esf* filter exceed the *bogo*'s and the *tftS3F*'s more.

Table 3. Performance Statistics of Email Spam Filtering

	Time (sec)	1-ROCA (%)	TREC 2007 Rank
<i>esf</i>	2,834	0.0055	
<i>wat3</i>		0.0055	1
<i>tftS3F</i>	62,554	0.0093	2
<i>bogo</i>	25,100	0.1558	

In the experiment B, we run the *esf* filter under different random sampling rate R_{rs} from the 90% down to the 10%. The *esf* filter repeatedly runs 30 times for each random sampling rate, and here reports the mean performance among the 30 results for each random sampling rate. The detailed random sampling rate (R_{rs}), final indexed token compressing rate (R_{tc}), and performances are showed in Table 4. Where, the R_{rs} is a predefined priori value, while the R_{tc} is a posteriori value after the filtering task, and is defined as the ratio of the number of tokens in the final TLM to the total number of processed tokens during the filtering task. The space is the number of tokens in the final TLM storage.

Table 4. Random Sampling Rate, Token Compressing Rate and Performances

R_{rs}	R_{tc}	Time (sec)	Space (num)	1-ROCA (%)
100	100	2,834	15,754,699	0.0055
90	94	2,715	14,763,087	0.0055
80	87	2,607	13,660,951	0.0054
70	79	2,481	12,511,131	0.0053
60	71	2,139	11,257,499	0.0053
50	63	2,130	9,895,697	0.0055
40	54	2,094	8,467,245	0.0053
30	44	2,066	6,860,210	0.0055
20	32	2,028	5,071,819	0.0064
10	19	2,006	2,984,139	0.0066

We find that the 1-ROCA is almost a constant (≈ 0.0055) while the R_{rs} varying from the 100% down to the 30%, which indicates if we randomly remove up to 70% tokens at the time of online training, the 1-ROCA will not be influenced obviously. On average of the 30 results, there are four 1-ROCA values exceed the best one (0.0055). Table 4 shows that the final indexed token compressing rate approximates a direct ratio of the random sampling rate, which proves that random-sampling-based token feature selection according to the theoretical uselessness rate heuristic between 63% and 82% is effective in the online situation.

In the experiment C, the *wdc* classifier runs on the 12-category Chinese WDCT, and sets its random sampling rate $R_{rs}=1$. Through evenly splitting the TanCorp-12 dataset, we make the three-fold cross validation. The mean MacroF1 and the mean MicroF1 are showed in Table 5, where the results of other four classifiers are cited from existing researches [2]. The results show that the *wdc* classifier can complete classifying task in high MacroF1 (0.8696) and high MicroF1 (0.9126), whose performance exceeds the *centroid*'s, the *kNN*'s, the *winnow*'s, and approaches to the best *SVM* classifier's MacroF1 (0.9172) and MicroF1 (0.9483).

Table 5. MacroF1 and MicroF1 Results

	MacroF1	MicroF1
<i>SVM</i>	0.9172	0.9483
<i>wdc</i>	0.8696	0.9126
<i>centroid</i>	0.8632	0.9053
<i>kNN</i>	0.8478	0.9035
<i>winnow</i>	0.7587	0.8645

In the experiment D, we run the *wdc* classifier under different random sampling rate R_{rs} from the 90% down to the 10%. The *wdc* classifier repeatedly runs 30 times for each random sampling rate, and here reports the mean performance among the 30 results for each random sampling rate. The detailed random sampling rate (R_{rs}), training indexed token compressing rate (R_{tc}), and performances are showed in Table 6. Where, the R_{tc} is a posteriori value after the training, and is defined as the ratio of the token number in the TLM after the training to the total number of processed tokens during the training. On average of the 30 results, Table 6 shows that the training indexed token compressing rate approximates a direct ratio of the random sampling rate, which proves that random-sampling-based token feature selection according to the theoretical uselessness rate heuristic between 63% and 79% is also effective in the multi-category situation.

Table 6. Random Sampling Rate, Token Compressing Rate and Performances

R_{rs}	R_{tc}	MacroF1	MicroF1
100	100	0.8696	0.9126
90	93	0.8715	0.9136
80	86	0.8677	0.9119
70	79	0.8663	0.9113
60	71	0.8657	0.9114
50	63	0.8653	0.9103
40	53	0.8609	0.9083
30	43	0.8570	0.9051
20	32	0.8517	0.9010
10	19	0.8345	0.8921

6 Conclusion

This paper investigates the power law distribution and proposes a RSEB algorithm for the online multi-category TC problem. The experimental results show that the RSEB algorithm can obtain a comparable performance compared with other advanced machine learning TC algorithms in email spam filtering application and Chinese web document classifying application. The RSEB algorithm can achieve the state-of-the-art performance at greatly reduced space-time complexity, which can satisfy the restriction of limited space and time for many practical large-scale applications.

With the development of mobile computing and network communicating, the spam concept extends from email spam to instant messaging spam, short message service spam, and so on. A web document may belong to the hierarchical category or may

have multiple category labels. Further research will concern the short message TC problem, the multi-hierarchy TC problem, and the multi-category multi-label TC problem. According to the ubiquitous power law, the RSEB algorithm is more general and can be easily transferred to other TC applications.

References

1. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* 34(1), 1–47 (2002)
2. Tan, S., Cheng, X., Ghanem, M., Wang, B., Xu, H.: A novel refinement approach for text categorization. In: *CIKM 2005: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pp. 469–476 (2005)
3. Sculley, D., Wachman, G.M.: Relaxed online SVMs for spam filtering. In: *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 415–422 (2007)
4. Cormack, G.V.: Email spam filtering: a systematic review. *Foundations and Trends in Information Retrieval* 1(4), 335–455 (2008)
5. Han, E.-H(S.), Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) *PKDD 2000. LNCS (LNAI)*, vol. 1910, pp. 424–431. Springer, Heidelberg (2000)
6. Zhang, T.: Regularized winnow methods. In: *Advances in Neural Information Processing Systems*, vol. 13, pp. 703–709 (2000)
7. Liu, W., Wang, T.: Online active multi-field learning for efficient email spam filtering. *Knowledge and Information Systems* 33(1), 117–136 (2012)
8. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) *MCS 2000. LNCS*, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
9. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. *SIAM Review* 51, 661–703 (2009)
10. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. *IEEE Transactions on Neural Networks* 10(5), 1048–1054 (1999)
11. Liu, W., Wang, T.: Utilizing multi-field text features for efficient email spam filtering. *International Journal of Computational Intelligence Systems* 5(3), 505–518 (2012)
12. Zobel, J., Moffat, A.: Inverted files for text search engines. *ACM Computing Surveys* 38(2), Article 6 (2006)

Natural Language Understanding for Grading Essay Questions in Persian Language

Iman Mokhtari-Fard

Department of Computer Engineering
University of Jahad
Shahrekord, Iran
iman@jdchb.ac.ir

Abstract. many intelligent systems are intended to communicate with users through natural language. Understanding the natural language by the computer is one of the most essential operations in natural language processing. One of the applications of natural languages is in the exams having essay questions. The objective of this paper is to propose a method for designing an examiner machine and creating an intelligent evaluator to grade the users' given answers to the essay questions. Algorithms such as "phrase structure" are weak at natural language processing in "free word order languages" such as Persian. The recommended method in this paper is based on "dependency grammar" and is applicable for various natural languages. Reduction in evaluation time and increase in the accuracy are advantages of the proposed method.

Keywords: Artificial Intelligent, Natural language understanding, Essay Questions Examiner.

1 Introduction

The ability to comprehend the natural language is one of the most valuable features of intelligent systems. Linguistic studies and designing computer algorithms are required for achieving this feature. Many studies have been conducted in this area but only the English language has been studied in most of the cases. English language is more feasible for designing algorithm compared to other languages such as German, Chinese, Persian, Arabic, etc due to its regulated structure and lack of free word order characteristic. Thus, it is essentially significant to have an algorithm which can be used for other languages as well as English.

The methods of representing the sentence grammar have played an important role in the advent of natural language processing algorithms. "Phrase structure" and "dependency grammar" are two principal methods for syntactic representation. Dependency grammar algorithm is applicable for free word order languages unlike the phrase structure method. Thus, we have applied dependency grammar and characteristics of Persian language for implementation of the algorithm.

In computer science, different methods can be applied for displaying the data. These methods include graph, Semantic networks, and use of objects. In this paper, we

will represent the natural language in the form of objects with the aid of dependency grammar.

A system capable of effectively understanding the natural language can be applied in different usages of natural language including summarization, text categorization, and text translation and so on. Following depiction of understanding procedure of natural language by the objects, this paper proposes a method for application in the intelligent evaluation system; the results of the recommended method are investigated in Persian language.

Syntactic representation methods are included in section 2; variety of applied questions in the evaluations will be presented in section 3. The proposed method is investigated in section 4. Section 5 deals with introducing the stages of converting texts into objects. Subsequently, answer acquisition and its evaluation procedures are analyzed in chapter 6. In chapter 7, besides introducing the Persian language, the results of the recommended method will be presented for the sample text in Persian language, and finally section 8 will incorporate the conclusions and future activities.

2 Syntactic Representation Methods

Syntactic representation methods are crucially influential in natural language processing. The texts are analyzed during the stages of syntactic representation of natural language and the meaning of the text is obtained subsequently [1]. Therefore, it is particularly significant to be familiar with different syntactic analysis method. Among the syntactic representation methods, dependency grammar and phrase structure algorithms are more remarkably important [2].

2.1 Phrase Structure

This approach is vitally important as one of the primary methods proposed by Chomsky [3] based on which the Context-free Grammar (CFG) is operated. Every sentence in context-free grammar system consists of several phrases and each phrase is composed of a set of words. These languages are also referred as “formal languages”. The grammar written based upon the formal language is called “Generative Grammar” [4].

Context-free grammars comprise a set of rules and symbols; the symbols, in turn, are divided into terminal and non-terminal symbols [2].

2.2 Grammar Representation Based on the Dependency

This method was innovated by Teniere [5] as a novel approach in modern linguistics. In this method, the syntactic structure comprises words which are related to each other through asymmetrical dual relationships [6]; these relationships are called “dependency”. In this representation method, it is emphasized that every sentence has a central verb, and the sentence structure can be determined using the central verb and the type and number of its mandatory and optional complements. In other words, this method rejects the former procedures which used to emphasize on division of the

sentences into subject and predicate [7]. Each verb imposes specific states of dependencies; the syntactic capacity is one of the most significant concepts in the dependency grammar. The fundamental structure of a sentence is determined based on its central verb [8].

2.3 Comparison of Methods

One of the major differences between the dependency and generative methods arises from the value or position they consider for the subject of the sentence. The generative grammar, from the beginning, divides the sentence into main parts of noun phrase (subject) and verb phrase (predicate); it actually follows the Aristotle logic of sentence analysis regarding the subject as one of the principal parts of the sentence just like the verb. Engel believes [9] the sentence division into subject and predicate mainly exhibits the information structure and the distribution of the new or old data rather than representing the syntactic structure of the sentence.

On the other hand, the phrase structure grammar is not suitable for languages having free constituent order. The dependency grammar is an appropriate candidate for this purpose due to its structure [2].

3 Evaluation System of Users' Answers

The evaluation systems are able to ask the questions from users in different ways. The different approaches for posing the question comprise following methods [10]:

- Multiple-Choice Items

Multiple-choice items include a set of responses. For each question, you must choose the best response.

- True/False Items

With true/false items, you are given a sentence to read, and you must decide whether the sentence is true or false.

- Fill-in-the-Blank

Fill-in-the-blank items are sentences with key words missing. You must fill in the missing word or phrase. You may be given a list of answer words to choose from.

- Short-Answer Questions

Short-answer questions are items that are answered by writing a few words or sentences.

- Essay Questions

An essay is writing that you do in response to a question or prompt. Essay questions test whether you have a deep understanding of your subject.

For Evaluating short-answer or essay questions, the natural language processing is required due to necessity of analysis. Many examinations including TOEFL and PhD entrance exams in many countries include essay questions. Furthermore, contests with essay questions can be held via SMS in which it may take a lot of time to evaluate all the answers because of the large number of participants; human errors might also occur in these evaluations. It is very effective to enjoy a system having the ability to investigate the answers intelligently and calculate the grade acquired for every person.

4 Recommended Method

In the proposed algorithm, the questions and correct answers are separately received by the system in the form of natural language. Then, the accurate answer for each question is converted into objects which represent the object-based representations of the input texts. In the subsequent step, the user inputs the relevant answer for each question through GUI¹. The system converts the input text for each question into distinctive objects. For analyzing the accuracy level of the answers, the created objects are compared with each other and the answering grades are calculated.

The procedures of designing the stages were carried out in a way that can be used for different natural languages; in other words, not applicable only for a specific language. However, in the conducted researches, the procedures were studied only for Persian language which assumes a free word order.

All stages of the system have been illustrated in figure 1; the procedures start from data input in the form of natural language and continue up to analyzing the answers and calculating the users' grades:

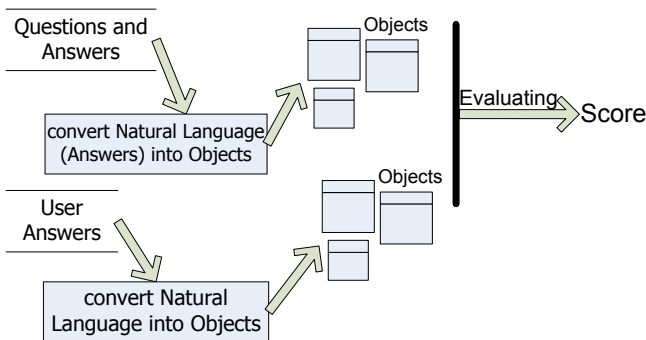


Fig. 1. System stages from receiving the questions and correct answers in natural language up to the evaluation of users' answers and calculation of grades

According to above figure, the recommended system consists of two main phases; the questions and correct answers are received in the first phase; then, the users' answers are acquired, evaluated and graded in the second phase.

5 Procedures of Converting Texts into Objects

We have used objects for understanding the natural language in the proposed method. Accordingly, each time a distinctive layer yields the object-based structure by acquisition of natural language. In the object-based representation, several objects might be obtained from the input text, and some relations are established between the objects whenever necessary. For converting the input text into objects, the stages are conducted as illustrated in figure 2.

¹ Graphic User Interface.

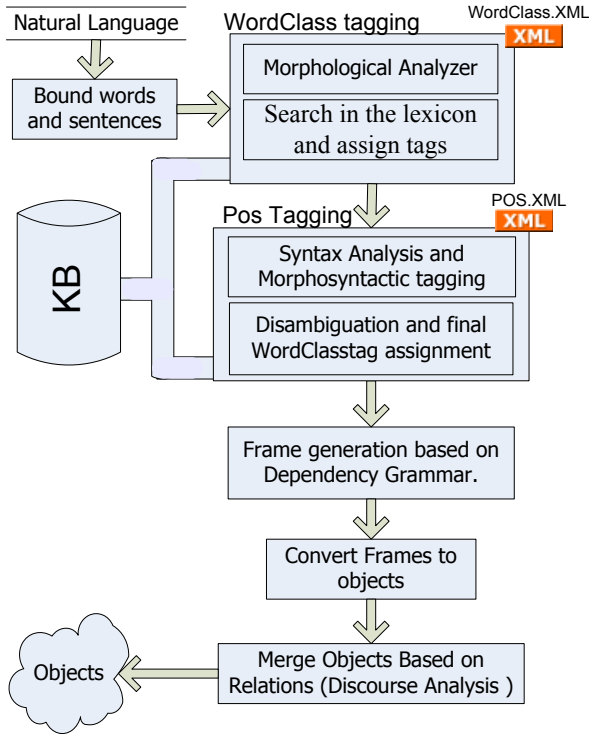


Fig. 2. Procedures of converting texts into objects

The stages illustrated in the figure are performed as follows:

5.1 Generation of Initial Objects

5.1.1 Word Class and Pos Tagging

Initially, the whole word (with affixes²) is searched in the database. It must be noted that only the word stem has been stored so as to reduce the size of the database. In the second part, based on the linguistic rules, the word is investigated in terms of having affixes; it is then analyzed and assigned the appropriate tag if necessary.

In this stage, every word or phrase is assigned a tag based on the linguistic characteristics. If a word has been assigned several tags in word class tagging stage – for example the word “مردم” in Persian language, considering the multiple-meaning property, can have a tag as the word “مَرْدَم” (I died) and also another tag as the word “مَرْدُم” (people). Then, as this word is investigated simultaneously and in association with other words in this stage, the multiple-tag words can be disambiguated according to the word function in the sentence and the adjoining words in the text. The result of this stage is saved in Pos.xml file.

² Prefixes or suffixes.

5.1.2 Frame Generation

A frame is generated by separately tagging for each sentence. Each frame includes slots to be filled by sentence constituents. The frame slots vary for every sentence and depend on the verb of the sentence. Dependency grammar characteristics are applied in this part. For instance, the verb “eat” will need an “object” dependent and a “subject” dependent (What was eaten?) (Who ate it?). Thus, the frame of the verb “eat” has got two slots. For example, for the sentences “My name is Sanova. I work in the university”, two frames will be generated as indicated in figure 3:

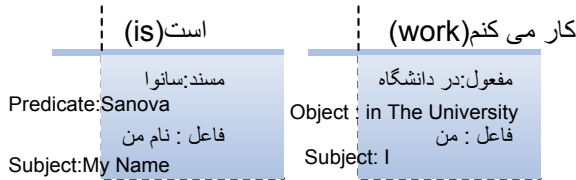
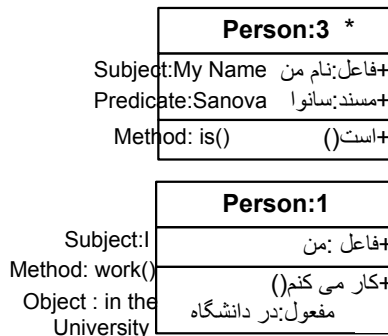


Fig. 3. The frames obtained from sample sentences

5.1.3 Converting Frames into Initial Objects

In Persian language, every sentence has a specific person based on its verb. This person can be “first person/ second person/ third person” singular or plural. In other words, there are totally 6 persons. Accordingly, an initial object is created for every frame. The object name is a number from 1 to 6 (1: for first person singular, 2: for second person singular, etc).

There exist some properties and a method inside the initial object. The method in the initial object is the same as the verb (action) of the sentence; the predicates are included as properties. The method can also have some dependents. For example, the initial objects for the frames generated in the previous part are created as shown in figure 4.



* The verb “است” (is) represents the first person singular in Persian language.

Fig. 4. The initial objects obtained from sample sentences

5.1.4 Generation of Final Objects

Presence of references as pronouns and relations between sentences necessitates us to combine the objects. In this stage, several objects may form an object, i.e. all the properties and methods pertaining to a subject are merged into an object. Moreover, an object may be converted into several objects. For instance: two objects of the previous example are converted into the two objects displayed in figure 5.

	Person:1
Subject:I	+فاعل:من
Name:Sandva	+نام:سانوا
Method: work()	+کار می کنم()
Object : in the University	+مفعول:در دانشگاه

Fig. 5. The final objects obtained from sample sentences

According to former discussions, the lexicological relations between words are stored in the database. Variety of object-based relations might be generated in the objects combination stage; e.g. an object may inherit from another object.

6 Receiving the Answers and Evaluation

The users' answers are separately received in the form of natural language. They are converted again into objects for evaluation. For this purpose, the module explained in section 5 is applied. The objects obtained from users' answers will be compared with objects generated from correct answers allowing the users' grades to be evaluated.

For certain questions, the order of answer parts is important; therefore, at the time of generation of object from the correct answer, each part is assigned an identifier which determines the order. The sensitivity to answer is also activated for that question.

Sometimes mentioning all the items is required for gaining the whole grade in questions having answers in the form of lists of items, or the grade is divided so long as some of the items are responded. These facts are also stored when generating the objects from the correct answers for every question.

7 Persian Language and Investigation of System Stages

Persian, also known as Farsi or Parsi, is an Indo-European language spoken and written primarily in Iran, Tajikistan and parts of Afghanistan. Persian alphabet contains 32 letters. Persian is written from right to left. Some other languages like Arabic, Kurdish, and Urdu use Persian's form of penmanship but have their own specifications. Persian also has its own specifications such as not using accents (except in special cases) and polymorphism in writing. The language has remained remarkably stable since the eighth century. It has a subject-object-verb word order, but has some head-initial structures [11].

Based on the dependency grammar, there are a number of dependents for any verb in every language indicating the capacity of that verb. For example, in [8, 12, 13, 7] the verb dependents have been determined for English, Chinese, German and Persian languages respectively. The current discussions are based upon the verb dependents in

Persian language. Considering the free word order structure of Persian language, every sentence can be written in several forms while all the forms convey the same meaning. We generate the same object from different forms of a sentence with the aid of the objects generated by dependency grammar.

In the implementation for Persian language, a text is initially received by the user interface. The determination of the boundaries of words and sentences below the first phase sub modules, class tagging is performed for each word. The results of this stage are saved in a XML file. For example, if the answer for the question “What does Sanova do?” is the sentences “She goes to a journey. She travels with bus”; then, according to the correct answer, the wordclass.xml is firstly created, and the tagging operation is subsequently done in sentence level by performing syntactic processing on this file resulting in the creation of pos.xml file as indicated in figure 6.

```

<text>
  <sentence1>
    <subject>سانووا</subject>
    <object>به سفر</object>
    <verb>می رود</verb>
  </sentence1>
  <sentence2>
    <subject>او</subject>
    <object>با اتوبوس</object>
    <verb>مسافرت می کند</verb>
  </sentence2>
</text>

```

Fig. 6. The structure of POS.xml file

The frames of each sentence are generated using the XML files and then the corresponding objects of the frames are created. The created frames and objects have been illustrated in figure 7.

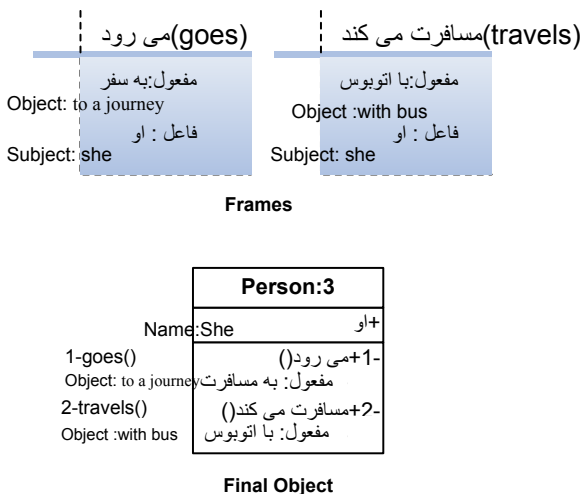


Fig. 7. The structure of the created frames and objects

When the set of objects was generated for each correct answer, the examinee inputs the relevant answer to each question via the user interface; the user's answer is in turn converted to a relevant object. For example, if in the answer to the previous question the user replies "Sanova goes to a journey"; following the creation of respective XML files, the object demonstrated in figure 8 is created.

Person:3	
Name:Sanova	+سانوا
1-goes()	-1+می رود()
Object: to a journey	مفعول: به مسافرت

Fig. 8. The object generated from user's answer

The object generated from user's answer is compared with the object created from the correct answer. The two objects may not be completely identical but they might have common methods or attributes for which the user's grade must be evaluated based on the discussions in section 6.

8 Conclusion and Future Work

Understanding the natural language by computers is one of the important matters in the area of information recovery systems. Many phrase structure algorithms have been proposed for natural language processing. The discrepancy of phrase structure system in the free word order languages revealed the significance of using dependency grammar algorithms.

Applying dependency grammar and conversion of natural language into objects, the computers can have a better comprehension of the natural language. The results of understanding the natural language can be used in different applications such as text summarization, text categorization, text translation systems and so on. We applied the system in the evaluation and grading system of essay questions in order to manifest the capability of the proposed method.

The complexity of natural language processing and the phrase structure method of the natural language processing have resulted in the absence of considerable researches in the field of devising machines for evaluation of users' answers to essay questions. Using the Recommended method, a better comprehension of the natural language can be provided for the computers.

Acknowledgements. The author thanks Dr. Omid Tabibzadeh and Marzie Badiie for explaining all the tricky details of the Persian Dependency grammar.

References

- [1] Canne, R., Kempson, R., Marten, L.: The Dynamics of Language. Elsevier Academic Press (2005)
- [2] Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edn. Pearson, Prentice Hall, Upper Saddle River, New Jersey (2009)

- [3] Chomsky, N.: *The Logical Structure of Linguistic Theory*. Plenum (1956)
- [4] Lester, M., Beason, L.: *The McGraw-hill handbook of English grammar and usage*, 1st edn. McGraw-Hill, New York (2005)
- [5] Tesnière, L.: *Esquisse d'une Syntaxe structurale*. Klincksieck, Paris (1953)
- [6] Kübler, S., McDonald, R., Nivre, J.: *Dependency Parsing*. Morgan & Claypool (2009)
- [7] TabibZadeh, O.: *Verb Valency and Basic Sentence Structures in Modern Persian (A Dependency-Based Approach)*. Nashr-e Markaz Publishing Co., Tehran (2006)
- [8] Allerton, D.J.: *Valency and the English Verb*. Academic Press, London (1982)
- [9] Engel, U.: *Kurze Grammatik der deutschen Sprache*. Iudicium Verlage, München (2002)
- [10] DesMarais, R.: *Student Success Handbook*. New Readers Press (2008)
- [11] Iranpour Mobarakeh, M., Minaei-Bidgoli, B.: *Verb Detection in Persian Corpus*. *International Journal of Digital Content Technology and its Applications* 3 (March 2009)
- [12] Li, W.: *A Dependency Syntax of Contemporary Chinese*. Institute of Linguistics, Chinese Academy of Social Sciences, manuscript (1989)
- [13] Fischer, K.: *German-English Verb Valency: A Contrastive Analysis*. Gunther Narr, Tübingen (1997)

Learning to Extract Attribute Values from a Search Engine with Few Examples^{*}

Xingxing Zhang, Tao Ge, and Zhifang Sui

Key Laboratory of Computational Linguistics (Peking University), Ministry of Education
{ zhangxingxing, getao, szf }@pku.edu.cn

Abstract. We propose an attribute value extraction method based on analysing snippets from a search engine. First, a pattern based detector is applied to locate the candidate attribute values in snippets. Then a classifier is used to predict whether a candidate value is correct. To train such a classifier, only very few annotated <entity, attribute, value> triples are needed, and sufficient training data can be generated automatically by matching these triples back to snippets and titles. Finally, as a correct value may appear in multiple snippets, to exploit such redundant information, all the individual predictions are assembled together by voting. Experiments on both Chinese and English corpora in the celebrity domain demonstrate the effectiveness of our method: with only 15 annotated <entity, attribute, value> triples, 7 of 12 attributes' precisions are over 85%; Compared to a state-of-the-art method, 11 of 12 attributes have improvements.

1 Introduction

One of the most important goals of building knowledge bases is to build a big table of *entities* with *attributes* and the corresponding *attribute values*. For example, for a celebrity, has the *attributes*: “height”, “weight”, and “education” and their corresponding *attribute values*: “1.75m”, “65kg”, and “xxx university”, etc. Entity extraction ([9, 13]) and attribute extraction ([10, 12]) are relative mature technologies, and this paper focuses on the more challenging task, attribute value extraction. The problem setting is: given a list of named entities of a domain (e.g. celebrities), and the attribute names (e.g. birthdate, height, etc.), an algorithm is expected to output triples of <entity name, attribute name, attribute value> for all the entities and attributes. In this paper, when we mention “value”, if not specified, it means “attribute value”.

An important observation is that many entity attribute values can be found in free text, e.g., there are many sentences like “Chris Pine received his bachelor’s degree from University of California, Berkeley” from which we can extract the “education” attribute value. Hence it is feasible to extract attribute values from the web text. However, the challenges are obvious:

- (A). It is a huge computational obstacle to parse all the free text in the entire web;
- (B). How to make certain that a piece of text is indeed description of the target attribute values other than some irrelevant information.

^{*} This paper is supported by NSFC Project 61075067 and National Key Technology R&D Program (No: 2011BAH10B04-03).

In our approach, for a target entity and the attribute name, we take the following steps to get the attribute value: **Step1**: we formulate queries according to the entity and the attribute, and submit them to a search engine and harvest the returned snippets, which often contain candidate attribute values. **Step2**: a pattern based detector is applied to locate the candidate attribute values in each snippet. **Step3**: a statistical classifier is used to predict whether a candidate value in **Step2** is a correct one. **Step4**: as a correct value often appears in multiple snippets, to exploit such redundant information, all the individual predictions of **Step3** are assembled together by voting and then output the final answer of the attribute value.

In the above steps, the key issue is how to train the classifiers in **Step3** and determine the voting weights in **Step4**. Directly labelling the candidate slots in the snippets to train the classifiers for **Step3** requires tremendous human effort. Instead, we require few accurate (manually labelled or from some trustful knowledge bases) <entity, attribute, value> triples, and use again the search engine to find large amount of the pieces of texts containing these correct attribute values. Thus they can be used as pseudo labelled data to train the classifier of **Step3**.

In the proposed approach, challenge (A) is conquered by leveraging a search engine to help us find the candidate pieces of text about the entity attributes from the entire web. The underlying assumption is: if a powerful search engine cannot find the pages containing the correct attribute value for an entity, the entity/attribute value must be very rare and we give up. Also, considering the current commercial search engines have already indexed more than billions of web pages, this assumption is reasonable. For challenge (B), our approach adopts a learning approach with very few labelled data. The underlying assumption is, for the same attribute, although different entities will have different attribute values, the expression ways in free text are similar. In addition, the assembling process in **Step4** well utilized the redundancy information from the web to depreciate noise: if many pages agree on an attribute value, it is more likely to be true.

We evaluated our algorithm in celebrity's domain (on both Chinese and English corpus), and remarkable performance is achieved: 7 of the 12 attributes' precision are over 85%. We also compared our algorithm with a state-of-the-art system and find that 11 of the 12 attributes have promising improvements.

The contributions of this paper are: (1) We proposed an approach to leverage a search engine to retrieve the candidate pieces of free texts describing a target entity and attributes from the entire web, rather than very limited websites like Wikipedia as in the previous approaches ([11, 14]). Experiments show that this can significantly improve recall. (2) We proposed a learning approach which learns the ways of describing attribute values of entities in free text. By utilizing the search engine again, the learning approach only requires very few labelled ground truth triples of <entity, attribute, value>.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 presents the proposed algorithm. The experimental setup is presented in Section 4, and the paper concludes in Section 5.

2 Related Work

The existing *attribute value extraction* methods can be roughly categorized into two types: rule-based and machine-learning-based algorithms. Rule based algorithms mainly rely on attribute specific rules. [6] is the champion team of the attribute value extraction task in *WePS* 2009. First, they classify the pages by checking keywords in page title; then for each type of page, keywords and rules for an attribute is employed to extract attribute values. [7] extracts numerical values by sending queries such as “Obj is * [height unit] tall” to extract attribute values. These algorithms above need too many complex attribute specific rules, and they usually have high precisions but low recalls. However, the rules in our method are much simpler; thus, recall is guaranteed. Further, a robust statistical classifier is employed to confirm the extractions.

Machine learning based algorithms usually need some training data. [11] employs co-EM to extract <attribute, value> pairs from product descriptions. As an attribute may have several names (e.g. weight, strung weight), to build a knowledge base, we must disambiguate attribute names, which is not an easy task. [2] mainly extract numerical values; it is viewed as a decision variable that whether a candidate value should be assigned to an attribute. The final value of the decision variables are assigned by solving a constrained optimization problem and the ground facts in training data are constrains of the problem. To further increase the performance, some attribute specific common sense (e.g. unstrung weight is smaller than strung weight) is additionally introduced as constrains. In addition, the computational cost is great. While our algorithm can extract other values besides numerical values, and additional common sense constrains are not needed. The work in [14] is most related to ours. They present *Kylin*, which fills the empty values in Wikipedia’s infobox. *Kylin* matches the existing values in infobox back to wiki articles and train a sentence classifier to predict whether a sentence contains certain type of value. Then extracting attribute values from these selected sentences is viewed as a sequential labelling problem, in which CRF is employed. The training data is acquired again by leveraging the existing values in infobox. But they only extract attribute values from a single Wikipedia page, and do not take advantage of the information from other sites. In [16], they also noticed that information on a single page is inadequate, so they employ the ontology in [15] and then articles in hypernym and hyponym classes are used as additional training data. To increase recall, they also use a search engine to get some relevant pages and extract attribute values from these pages. But if models are trained in Wikipedia pages and it runs on general pages, the extraction results may suffer. Besides, for some multi-word values, CRF may cause boundary detection errors. While our method extracts the candidate values as a whole, and boundary detection errors will not happen.

The tasks of *Relation Extraction* and that of *Attribute Value Extraction* are similar. Some relations such as *bornOnDate* and *graduatedFrom* in *relation extraction* are just the attribute values of a person’s *birthdate* and *education* attributes, while some relations such as *producesProduct* and *publicationWritesAbout* cannot be viewed as attribute values of certain entity. Pattern-based relation extraction (e.g. [8], [17], [3]) usually bootstraps with some seed relations (facts), and in every iteration new patterns

and facts are extracted and then evaluated by statistical measures such as PMI. Usually, recalls of these methods are high, but these methods often produce noisy patterns and may drift away from target relations.

[5] proposed a multi-stage bootstrapping ontology construction algorithm. In each iteration, they integrated CPL and CSEAL ([4]), which are all pattern-based relation extractor, to fetch candidate facts, and then a statistical classifier is employed to further refine the meta-extractions. However, after each iteration of rule learning, bad extraction rules must be removed manually.

Our approach has a rule based component to detect the candidate attribute values in a piece of free text, getting high recall but low precision candidates. And we also have a successive learning component, a statistical classifier, to further confirm whether a candidate value is indeed a correct one considering the features from its context.

3 Methodology

Suppose that in the target domain, there are N unique named entities and A attributes, and our system is expected to output $N \times A$ attribute values. For each entity e and an attribute name a , we will get one attribute value v . The workflow of our approach is described in Figure 1. The system has four main successive components: *Corpus Fetcher*, *Candidate Value Detector*, *Attribute Value Classifier* and *Voter*. First, in the component of *Corpus Fetcher*, we formulate a query by concatenating the entity e and the attribute name a . For example, for an entity $e = \text{“Michael Jackson”}$, and the attribute $a = \text{“Birthday”}$, the formulated query is *“Michael Jackson Birthday”*. The query is sent to a search engine, and we fetch the titles and snippets of the top K ($=25$ in our experiments) returned results. Then, entering into the component *Candidate Value Detector*, for an attribute, we define some patterns to filter the obviously wrong slots in a snippet, for example, some attribute values like *nationality* must belong to a finite enumerable list. Such a pattern based detector is used to roughly locate candidate values in a snippet. Notice that this detector will have high recall but low precision. Next is the task of the component of *Attribute Value Classifier*, a binary statistical classifier that is used to predict whether a candidate value is confident. The prediction is based on features extracted from snippets containing the candidate value

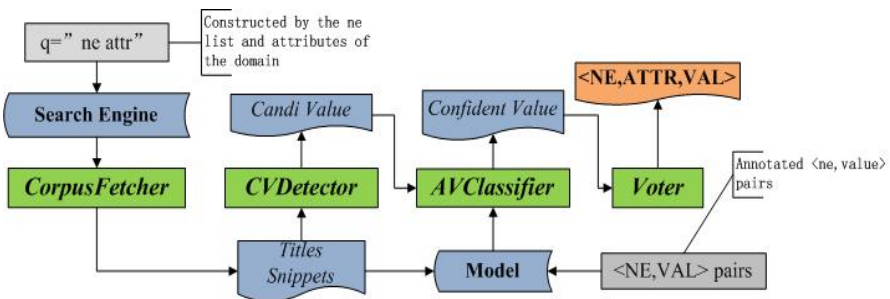


Fig. 1. System architecture. The grey parts are the inputs of the system.

and the snippet’s corresponding title. Then all the non-confident candidate values are discarded. Finally, notice that the previous step may produce the same candidate value in different snippets, and it is the often case that the true attribute value appears in multiple snippets. To utilize this redundancy information and make the final decision, we adopt a component of *Voter* to assemble the predictions in the previous step by voting. The voter assigns a weight to each confident value (the refined candidate value in the last step) and accumulates the weights of the same attribute value as its voting score. The candidate values are ranked by their voting score and the extracted attribute value is the one with the highest score. For instance, we have four confident values (three unique ones), “v1 v2 v1 v3”. After weighting, each value gets a weight, e.g. “v1:0.98 v2:0.64 v1:0.72 v3:0.99”. After accumulation, the voting scores are “v1:1.7 v2:0.64 v3:0.99”. So, the final extracted attribute value is “v1”.

In the following parts of the section, we will introduce the main components of the system in detail.

3.1 Candidate Value Detector

For an attribute a , there should be a *validity checker* to judge whether a candidate value is a valid one. For example, for the attribute *birthday*, a candidate value should be of a valid date format. This paper considers the following two broad cases where the validity checker is easy to obtain: (1) the range of an attribute value is a finite enumerable set, e.g., a valid nationality value must belongs to the set of names of all the countries in the world (there are overall 192 countries); (2) the range of an attribute value can be described by nontrivial regular expressions. For example, birthday values have such formats as “0000-00-00” (e.g. 1986-10-12) or “<Month> 00, 0000”, etc. The tested attributes in the paper are all of the two cases, whose valid formats (validity checkers) are shown in Table 1. Actually, for other cases where a value validity checker is provided, the method of this paper can also apply. For example, for the attribute of *spouse*, whose value should be of the type of *person*, we can define a validity checker based on some NER algorithms.

Table 1. Attributes and their formats

Attribute	Format	Attribute	Format
体重 (weight)	(?i)(\d+(\.\d+)?)\s?(kg 千 克 公斤 磅)	出生日期 (birthdate)	(\d+)(\d+)(\d+)年(\d+)(\d+)(\d+)月(\d+)(\d+)日, (\d+)-(\d+)-(\d+), ...
国籍 (nationality)	Entities in Country list	毕业院校 (education)	Entities in School list
民族(Ethnic Group)	Entities in Ethnic Group list	英文名 (English name)	[A-Z][A-Za-z]+(\s[A-Z][A- Za-z]+(\s[A-Z][A-Za-z]+)?)?
血型(blood type)	(A B AB O 0)\s?型	身高 (height)	(?i)(\d+)\s?(cm 厘米), (\d+(\.\d+)\s?)米
birthdate	(\d+)\s+(January Jan Febru ary Feb ...)\s+(\d+), (\d+)-(\d+)-(\d+), ...	height	(?i)(\d+)\s+?cm, (?i)(\d+)\s*ft\s*(\d+)\s*in, (?i)(\d+(\.\d+)\s+)\s+m, ...
nationality	Entities in nationality list	weight	(?is)(\d+(\.\d+)?)\s+?kg, (?is)(\d+(\.\d+)?)\s+?lb

The validity checker for an attribute plays the role of a candidate value detector, which detects the valid values in the snippets as candidates. Notice that in this step, some valid but incorrect values may also be extracted as candidates. For example, when we detect valid birthdate values, some irrelevant dates such as the report dates and the page’s dates may also be extracted. Actually in this step, we care more about recalls than precisions. In the next steps (Section 3.2, and Section 3.3), from different aspects, we will further filter the candidate values produced in this step to promise high precision. Section 3.2 will filter the candidate values by a classifier considering the context of a candidate value in a snippet. And Section 3.3 will utilize the fact that a correct value often appears in multiple snippets to design a voting mechanism, so that the correct value agreed by multiple snippets is picked up while many incorrect candidates are filtered.

3.2 Attribute Value Classifier

The candidates *Candidate Value Detector* output may be incorrect for the target entity. It may be the case that one snippet may describe several named entities (e.g. celebrities), and the candidate may be other entity’s value. In addition, a candidate may be some noise in snippets. For example, the candidate “birthdate” may be just the report date of a piece of news. Thus, we introduce a statistical classifier, the *Attribute Value Classifier*, which aims at refining these candidates by utilizing the features in the snippet containing the candidate and the corresponding title of the snippet.

We train one binary classifier for each attribute, which tries to predict whether a candidate value is confident. The classification model we used is Maximum Entropy Model, which can provide the probabilities of predictions. And in the next section, these probabilities will be used to improve voting.

• Features

The prediction is based on features extracted from the snippet containing the candidate value and the corresponding title. We use two types of features: title features describing topics of search results, and snippet features encoding local information of search results. Feature (1)-(3) are title features, and (4)-(7) are snippet features. All these features except (3) and (7) are binary.

- (1) Whether the title contains the current named entity. This is a strong indication that the search result is describing the current named entity.
- (2) Whether the title contains other named entities of the same class. For example, when extracting Michael Jackson’s birthdate, we will see if other celebrities’ names are in the title. This is a strong indication that the search result is describing other named entities or the current and other named entities at the same time.
- (3) Other words with their POS tags in the title. For example, the title is “Michael Jackson - Wikipedia, the free encyclopedia” and the current named entity is “Michael Jackson”. Then feature (3) is “-/: Wikipedia/NNP ./, the/DT free/JJ encyclopedia/NN”.
- (4) In the sentence that the candidate value appears, whether the current attribute name appears. This is a strong indication that the candidate value is the value of the current attribute.

- (5) In the sentence that the candidate value appears, whether the current named entity appears. This is a strong indication that the candidate value is related to the current named entity.
- (6) In the sentence that the candidate value appears, whether the other named entities (of the same class) appear. This indicates that the snippet is describing other named entities, and thus the candidate value may not be a confident value.
- (7) Other words with their POS tags and distance to the candidate value in the sentence. For example, in the sentence “Michael Jackson was born on August 29, 1958”, “Michael Jackson” is the current named entity and “August 29, 1958” is the candidate value. Then feature (7) is “was/VBD/-3 born/VBN/-2 on/IN/-1”. Note that distances of the words on the candidate value’s left are negative and distances of the words on the candidate value’s right are positive.

The intuitions under the feature design are: (1) if a search result is describing the current named entity, then it is likely that the candidate value in the snippet is correct; (2) if the current named entity or the current attribute appears in the same sentence with the candidate value, then it is also likely that the value is correct.

• Generating Training Data

Training the classifier needs labelled data. It is not practical to manually annotate the correct attribute values in each snippet. We propose a method to reduce the labelling effort. Rather than relying on direct annotations on each snippet, we only require a few correct <entity, attribute, value> triples, which are matched back to the search results (title and snippets) to generate training data for the classifier. An advantage of this method is that it is easy to get a few correct <entity, attribute, value> triples, either by human labelling or from some structured sites such as Wikipedia.

Table 2. Numbers of training examples generated by 15 <entity, attribute, value> triples for each attribute. “pos” means the proportion of positive training examples.

Attribute	N	pos	Attribute	N	pos	Attribute	N	pos
体重 (weight)	245	53%	出生日期 (birthdate)	1181	18%	国籍 (nationality)	419	74%
毕业院校 (education)	307	68%	民族 (Ethnic Group)	82	89%	英文名 (English name)	74	24%
血型 (blood type)	188	68%	身高 (height)	207	60%	birthdate	393	34%
height	77	41%	nationality	338	62%	weight	77	74%

Step 1: Annotate some <entity, attribute, value> triples (only 15 triples in the experiment). We can also get these triples from some structured sites (e.g. Wikipedia, Bio27, etc.).

Step 2: Submit queries to a search engine and match these <entity, attribute, value> triples back to search results. For example, we are extracting celebrities’ birthdate, and we have a labelled triple of <‘Michale Jackson’, ‘Birthdate’, ‘19580829’>. We send the query “Michael Jackson birthdate” to a search engine and get the top K

($K=25$ in our experiments) search results (titles and snippets). Then the *Candidate Value Detector* extracts all the candidate values and converts them to standard formats ('yyyymmdd'). If a candidate value equals to the true value ("19580829"), then we annotate a positive label to the value in the snippet and get a positive training example; otherwise, we get a negative training example. In this way, each candidate value in a snippet will produce a training example. The number of training examples generated by the 15 <entity, attribute, value> triples for each attribute is shown in Table 2. The number of positive training examples is not necessarily less than the number of negative training examples. Proportions of positive training examples are in the 'pos' columns.

3.3 Voter

After the classification, there may be several candidate values for an entity's attribute and the correct value often appears in multiple snippets. Intuitively, the most frequent candidate value is most likely to be the correct value. Therefore, a simple strategy is to count how many times a candidate value is classified as confident value by the classifier. However, this may cause a problem when several candidate values get the same highest score. To alleviate the problem, we leverage the classification probabilities provided by the *Attribute Value Classifier* and use the probability as each vote's weight. Experiments show that this strategy can improve precisions and recalls by about 1%.

4 Experiment

We use Baidu¹ (Chinese) and Google (English) to test our algorithm. For our *Attribute Value Classifier*, we employ a Maximum Entropy model implemented by Le Zhang [18]. We employed L-BFGS and the Gaussian prior is 1.0.

Table 3. Numbers of nonempty values for each attribute

Attribute	N	Attribute	N	Attribute	N	Attribute	N
体重 (weight)	667	出生日期 (birthdate)	3004	国籍 (nationality)	3726	毕业院校 (education)	1162
民族 (Ethnic Group)	861	英文名 (English name)	2759	血型 (blood type)	1652	身高 (height)	2985
birthdate	1532	height	1543	nationality	1295	weight	1522

¹ <http://www.baidu.com>

² http://ent.qq.com/c/all_star.shtml

³ <http://app.ent.ifeng.com/star/>

⁴ <http://baike.baidu.com>

⁵ <http://www.hudong.com>

⁶ <http://www.wikipedia.org>

⁷ <http://www.bio27.com>

4.1 Evaluation Dataset

The experiments were conducted in the celebrity domain (on both Chinese and English corpus). We collected the 4476 celebrities in “qq entertainment²” as our Chinese named entity list. And we crawled celebrities’ data in “qq entertainment”, “ifeng entertainment³”, “baidu baike⁴”, “hudong baike⁵” and “Wikipedia⁶” as our Chinese standard evaluation dataset. Similarly, we collected 1600 celebrities in “bio27⁷” as our English named entity list. And we crawled celebrities’ data in “bio27” as our English standard evaluation dataset. Some named entities’ values cannot be found in all these sites, and their values are empty in our dataset. Numbers of nonempty values for each attribute are in Table 3.

4.2 Experimental Results

We tested 14 attribute values, and 8 of them are on Chinese corpora (alias Cx), while 4 of them are on English corpora (alias Ex). In addition, we use 15 <entity, attribute, value> triples for each attribute to generate training data and train the *Attribute Value Classifier*. We evaluated their Precisions and Recalls. As some named entities’ certain attribute may not exist (e.g. M. Jackson is an American, and does not have the attribute “民族(Ethnic Group)”), we only evaluate the attribute values in the our dataset.

The results in Table 4 show that 7 of the 12 attributes’ precision are over 85%.

Table 4. Results with 15 <entity, attribute, value> triples. In graphs, attributes are replaced by their “Alias”. “Cx” donates a Chinese attribute, while “Ex” donates an English attribute.

Attribute	Alias	P	R	Attribute	Alias	P	R
体重 (weight)	CW	0.7711	0.6312	出生日期 (birthdate)	CB	0.8582	0.7397
国籍 (nationality)	CN	0.9239	0.8341	毕业院校 (education)	CE	0.8512	0.7040
民族(Ethnic Group)	CEG	0.8357	0.6202	英文名 (English name)	CEN	0.6320	0.4937
血型(blood type)	CB	0.9322	0.7488	身高 (height)	CH	0.8676	0.6938
birthdate	EB	0.9139	0.9073	height	EH	0.7120	0.6137
nationality	EN	0.9451	0.9444	weight	EW	0.7533	0.6419

4.3 Single Site vs. Multiple Sites

To increase recall, we leverage a search engine to extract attribute values from the entire web, rather than very limited websites. In this section, we provide evidences for this claim. We studied the best recalls an algorithm can achieve on two sites, namely, Wikipedia and Baidu-baike (Chinese version Wikipedia), and compared them with that of the proposed algorithm. Specifically, for a target entity and one of its attributes, if its correct attribute value for the current attribute can be found on the entity’s page, it does count for a correct extraction. The comparison in Figure 2 shows

that the recall of the proposed algorithm is better than the best recalls on Wikipedia and Baidu-baike. That is to say that no algorithm targeting at these two sites can have a better recall than the proposed algorithm. Therefore, it is necessary to leverage the information from multiple sites.

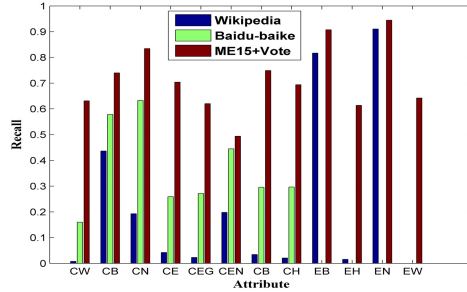


Fig. 2. Best recalls an algorithm can achieve on Wikipedia and Baidu-baike and the recall of the proposed algorithm. The attributes on x-axis are all their aliases (details in Table 4).

4.4 Comparison to a Previous System

In this section, we compare the performance of different methods on algorithm level (on the same corpora).

It can be difficult to compare our results with other attribute value extraction systems. Unlike semantic role labelling, there are some public available datasets (e.g. PropBank and Pen TreeBank). The datasets used by previous systems are different from ours. One feature of our system is the leverage of multiple sites data, so we cannot only use Wikipedia as in [14]; besides numerical values, we can extract other kind of values, so we cannot use the dataset in [2]. Finally, we implemented *Kylin* [14].

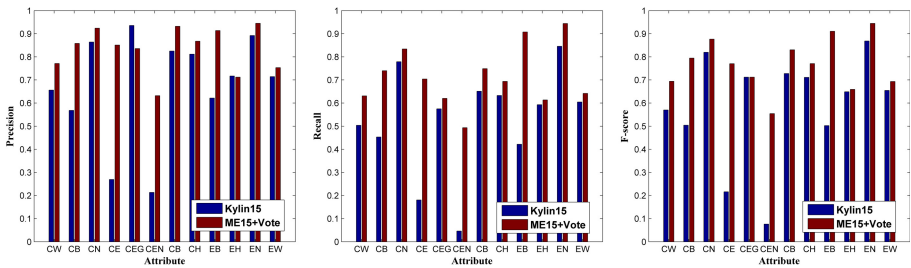


Fig. 3. The comparison between *Kylin* ([14]) and our method (ME15+Vote). Both systems use 15 \langle entity, attribute, value \rangle triples. The attributes on x-axis are all their aliases (Table 4).

In experiments, we implemented *Kylin* in [14] to extract attribute values from snippets. We tested *Kylin* with 15 annotated \langle entity, attribute, value \rangle triples and 100 annotated \langle entity, attribute, value \rangle triples (the same amount of annotated triples with our method) respectively. And the comparisons are shown in Figure 3 and Figure 4 respectively.

In Figure 3, ME15+Vote have a better precision, recall and F-score on all attributes except the attribute “民族(Ethnic Group)”(CEG). Among these attributes, the improvements on “出生日期(birthdate)”(CB), “毕业院校(education)”(CE), “英文名(English name)”(CEN) and “birthdate”(CB) are obvious. In Figure 4, results of *Kylin* with 100 annotated pairs are better than their results with 15 annotated triples on most attributes. However, the results are similar with that of 15 annotated triples: still only the results of “民族(Ethnic Group)”(CEG) are better than the proposed method. The proposed method leverages a rule-based detector to locate the candidate values and during classification, the features in title, which reflects the topic information of the snippet, are used. We believe these factors above lead to a better performance.

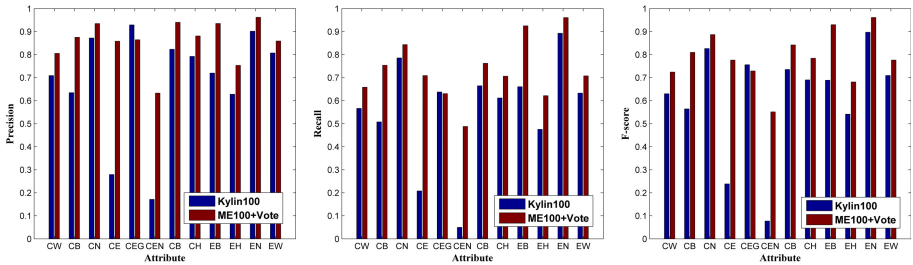


Fig. 4. The comparison between *Kylin* ([14]) and our method (ME100+Vote). Both systems use 100 <entity, attribute, value> triples. The attributes on x-axis are all their aliases (Table 4).

4.5 Impact of the Amount of Annotated Data

We also studied the effects of different amount annotated <entity, attribute, value> triples on F-scores. It is shown in Figure 5. We can see when the amount of annotated triples is between 5 and 15, some attributes’ (e.g. 体重(weight)) F-scores increase significantly; when the amount is between 15 and 100, F-scores do not increase much; when the amount is between 100 and 300, “英文名(English name)” and weight have about 5% increase. But for 5%’s improvement to annotate 20 times of data is not worthy. So we use 15 annotated triples, and they are sufficient for most attributes.

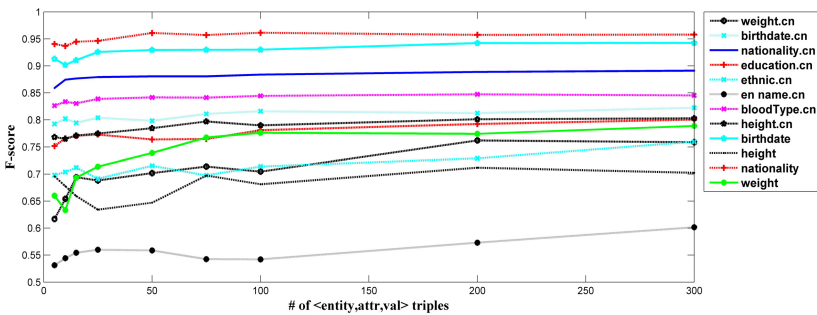


Fig. 5. The impact of amount of annotated <entity, attribute, value> triples on F-score

5 Conclusions

In this paper, we proposed an attribute extraction algorithm. The attribute values are extracted from snippets returned by a search engine. We use some strict (mainly rule based) methods to locate the possible values in the snippets. Then a classifier is used to predict whether the candidate value is a confident one. To train the classifier, we only need to annotate very little data, and sufficient training data will be generated automatically. A correct value may appear in multiple snippets, and we also have a strategy to vote and score the confident values.

References

1. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: IJCAI (2007)
2. Bakalov, A., Fuxman, A., Talukdar, P., Chakrabarti, S.: Scad: collective discovery of attribute values. In: Proceedings of WWW 2011, Hyderabad, India, pp. 447–456 (2011)
3. Cafarella, M.J.: Extracting and querying a comprehensive web database. In: CIDR (2009)
4. Carlson, A., Betteridge, J., Wang, R.C., Hruschka Jr., E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: Proc. of WSDM (2010a)
5. Carlson, A., et al.: Toward an architecture for never-ending language learning. In: Proceedings of AAAI 2010 (2010b)
6. Cimiano, P., Völker, J.: Text2Onto – a framework for ontology learning and data-driven change discovery. In: NLDB (2005)
7. Davidov, D., Rappoport, A.: Extraction and Approximation of Numerical Attributes from the Web. In: Proc. of ACL (2010)
8. Etzioni, O., et al.: Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.* 165(1) (2005)
9. Kozareva, Z., Riloff, E., Hovy, E.: Semantic class learning from the web with hyponym pattern linkage graphs. In: Proceedings of ACL 2008: HLT (2008)
10. Pasca, M., Van Durme, B.: Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs. In: Proceedings of ACL 2008, pp. 19–27 (2008)
11. Probst, K., Ghani, R., Krema, M., Fano, A., Liu, Y.: Semi-supervised learning of attribute-value pairs from product descriptions. In: IJCAI (2007)
12. Ravi, S., Pasca, M.: Using Structured Text for Large-Scale Attribute Extraction. In: Proceedings of CIKM 2008, pp. 1183–1192 (2008)
13. Wang, R.C., Cohen, W.W.: Language-independent set expansion of named entities using the web. In: ICDM, pp. 342–350. IEEE Computer Society (2007)
14. Wu, F., Weld, D.S.: Automatically semantifying Wikipedia. In: CIKM, pp. 41–50 (2007)
15. Wu, F., Weld, D.S.: Automatically refining the wikipedia infobox ontology. In: Proceedings of WWW 2008 (2008)
16. Wu, F., Hoffmann, R., Weld, D.S.: Information extraction from Wikipedia: Moving down the long tail. In: Proceedings of KDD (2008)
17. Xu, F., Uszkoreit, H., Li, H.: A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In: ACL (2007)
18. Zhang, L.: Maximum Entropy Modeling Toolkit for Python and C++ (2004), http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html

User-Characteristics Topic Model

Wenfeng Li, Xiaojie Wang, and Shaowei Jiang

Beijing University of Posts and Telecommunications

Abstract. This paper proposes a method to capture user's characteristics in a topic model frame, where user characteristics act as a latent variable that does not depend on texts. As it is obvious that different people possess different characteristics, they may perform differently even when they are facing the same document. These different characteristics can be showed as different views or different wording preference. We think this phenomenon has a great impact on modeling texts written or labelled by different people, especially on topic modeling. Experiments show that the model with user characteristics outperforms the original models and other similar topic models on corresponding tasks. A combination of the user's characteristics can not only provide better performance on normal topic modeling tasks, but also discover the user's characteristics.

Keywords: user characteristics, topic modeling, personalized model.

1 Introduction

With the development of Web2.0, users are becoming more and more deeply involved in the Internet, not only as readers, but also as authors. This development has made the quantity of text corpora on the Internet increase rapidly. As a result, it becomes more and more challenging to organize corpora efficiently, through this, users can find what they need conveniently. Dimensionality reduction is a reasonable way to model large amount of data and get short descriptions for texts which is useful for certain basic tasks such as classification or relevance judgments.

A vast number of statistical learning methods have been used to model the texts. Among them, a series of Latent Dirichlet Allocation (LDA) based topic models initiated by Blei[1] have been developed. LDA uses topics as latent variables for text description. It has been extended in several different ways and has achieved success in some applications. For example, supervised LDA [2] assumes that there is a label generated from each document's topic distribution. Labeled-LDA [3], TagLDA[4] and Multi-Multinomial LDA (MM-LDA)[5] have been used to model multi-labels text. Labeled-LDA constrains the topic distribution by user's labels as supervised information, while the tag set and the word set are assumed to be independently sampled from the document in TagLDA/MM-LDA. The Author-Topic Model (ATM)[6][7] models the interests and topics based relations of authors. The Author Interest Topic Model (AITM)[8] allows a number of possible latent variables to be associated with authors' interests, where the model assumes that each author has only one interest for one document.

Among these models, a basic assumption is all users (including authors and readers) have the same word distributions for a same topic. That is to say, when a topic is found, the word distribution of the topic is same for all users, no matter who they are.

But the fact is that when different people talk about the same topic, there is a big probability they will prefer to use different words. That is to say, for a same topic, there may be several different word distributions for different kind of people. For example, when two people talk about a mobile phone, one may first think about its capability of communication, while the other may first talk about its convenience, depending on their backgrounds and interests. When they talk about the convenience of the mobile phone, one may first use “portable”, another may first use “carry-on”, because of different wording preferences they may have. It is the same when different people read a document on the same topic, they will be concerned with different words in the document or use different words to tag the documents, depending on their backgrounds and/or wording preferences.

Based on the above observations, we assume word distributions for a topic is not only dependent on the topic, but also dependent on users. We assume there is a latent user characteristics for different groups for people, which makes different groups of users have different word distribution even for a same topic.

This paper aims to capture both latent topics and latent user characteristics in one LDA based model. Due to existing user characteristics, the word distribution on a similar topic for different users will be different. A topic model that does not concern these differences can be thought of as an average model of a large collection of different users with different characteristics. By making use of user-specific differences of topics, we aim to not only achieve better topic modeling for documents, but also extracting more information of both writers and readers(taggers) of the documents .

To combine the user’s characteristics in text modeling, we develop a user-characteristics LDA (UC-LDA). The difference between our model and previous is that our model assumes that words of a document are not only rested with document’s topic distribution, which is the same as that in LDA, but also controlled by the user’s characteristics distribution. Experimental results shown that our model (UC-LDA) outperforms LDA, ATM and AITM significantly in text modeling task. In addition, it can discover some interesting results about user’s characteristics which cannot be given by previous ones. Also, we applied the idea of user-characteristics (UC) to TagLDA (UC-TagLDA), from the experimental results, we can find UC is not just specific to a certain topic model, it can be applied to a wide range of topic models.

The organization of this paper is as followings: Section 2 describes how our model works with the example of UC-LDA and UC-TagLDA. The experimental results are shown in Section 3. Section 4 draws some conclusions.

2 User-Characteristics Topic Model

2.1 Motivation

Topic models like LDA only concern the generative process of the documents. For each document in the LDA model, the topic distribution θ_d is a multinomial distribution randomly sampled from a Dirichlet distribution, for each word in document d , the topic assignment $z_{di}^{(w)}$ is chosen from this topic distribution for the i th word, and then a word w_{di} is generated from a topic-specific multinomial distribution $\phi_{z_{di}}$.

As we have argued, when different people talk about the same topic, there is a great probability that they will prefer to use different words. A topic model that does not concern these differences can be thought as an average model of a large group of different users with different characteristics. We now add the user's characteristics to the generative process based on the following assumptions: When a word is chosen for a topic by a user, it not only rests with the topic, but also rests with that user's characteristics. Meanwhile, different types of characteristics generate different users.

These assumptions can be represented in graph shown in Figure 1(a).

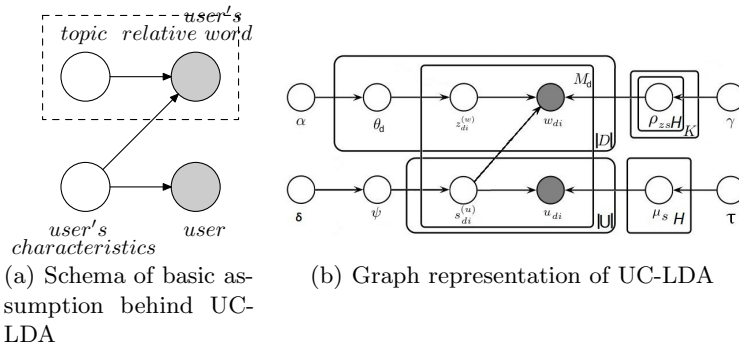


Fig. 1. Schema of basic assumption behind User Characteristics Topic Model

2.2 Description of User-Characteristics LDA(UC-LDA)

UC-LDA can be thought as a combination of the above assumptions and LDA. It gives a way to model users and documents at the same time.

With the combination of Figure 1(a) and LDA, we can get the graph representation of UC-LDA in Figure 1(b). In UC-LDA, each word is not only influenced by its topic assignment, but also by characteristics of the user who is relative to this word. Our notation is summarized in Table 1.

And the generative process of the User-Characteristics Topic Model is shown as follows:

1. Draw H multinomial μ_s from Dirichlet prior τ ;
2. Draw $K \times H$ multinomial ρ_{zs} from Dirichlet prior γ to represent the tag distribution, one for each topic z assigned to the characteristics s ;
3. Draw a multinomial ψ from Dirichlet prior δ ;
4. For each document d , draw a multinomial θ_d from a Dirichlet prior α ;
 - (a) For each word w_{di} and the user u_{di} who is relative to this word,
 - i. Draw a characteristics $s_{di}^{(u)}$ from multinomial ψ , and then draw a user from $\mu_{s_{di}^{(u)}}$
 - ii. Draw a topic $z_{di}^{(w)}$ from multinomial θ_d , and then draw a word from $\rho_{z_{di}^{(w)} s_{di}^{(u)}}$

Table 1. Notation used in our model

SYMBOL	DESCRIPTION
D, D , d	D is a collection of documents, $ D $ is the number of documents in the collection, and d is a document in the collection.
W, W , w	W is a collection of word tokens, $ W $ is the number of tokens in the collection, and w is a word in the collection.
T, T , t	T is a collection of tag tokens, $ T $ is the number of tokens in the collection, t is a token in the collection.
U, U , u	U is a collection of users, $ U $ is the number of users, and u is a user in the collection
w_{di}	the i th word in document d
t_{dj}	the j th tag in document d
u_{dj}	the user who give the j th tag in document d
K	number of topics
H	number of user's characteristics
N_d	number of words in document d
M_d	number of tags in document d
$z_{di}^{(w)}$	the topic assigned to the i th word in the document d
$z_{dj}^{(t)}$	the topic assigned to the j th tag in the document d
$s_{dj}^{(u)}$	the characteristic assigned to user who give the j th tag in the document d
θ_d	the topic distribution of document d
ψ	the characteristics distribution on the corpus
ϕ_z	the word distribution of topic z
ρ_{zs}	the tag distribution of topic z specific to characteristic s
μ_s	the user distribution of characteristic s

2.3 Inference

As a topic model can not be exactly inferred, we use Gibbs sampling to get an approximate inference of our model. For each iteration, we need to sample the topic of each word, and also need to sample the characteristics of the user who gives that tag.

The Gibbs sampling procedure can be seen in Figure 2. Where,

```

for each iteration :
  for  $d$  in  $D$ :
    for  $i = 1$  to  $N_d$ :
      draw  $z_{di}^{(w)}$  from  $p(z_{di}^{(w)}|\cdot)$ 
      draw  $s_{di}^{(u)}$  from  $p(s_{di}^{(u)}|\cdot)$ 
      update  $n(z_{di}^{(w)}, s_{di}^{(u)}, w_{di})$ ,  $n(s_{di}^{(u)}, u_{di})$  and  $n^{(w)}(d, z_{di}^{(w)})$ 
    end for
  end for
end for

```

Fig. 2. Gibbs sampling process of UC-LDA

$$p(z_{di}^{(w)} = z|\cdot) \propto \frac{n(z, s_{di}^{(u)}, w_{di})_{-t_{di}} + \gamma}{\sum_t (n(z, s_{di}^{(u)}, t) + \gamma) - 1} \times \frac{n^{(w)}(d, z)_{-w_{di}} + n^{(w)}(d, z) + \alpha}{\sum_d (N_d + M_d) + \alpha K - 1} \quad (1)$$

$$p(s_{di}^{(u)} = s|\cdot) \propto \frac{n(s, u_{di})_{-u_{di}} + \tau}{\sum_u (n(s, u) + \tau) - 1} \times \frac{\sum_u n(s, u_{di})_{-u_{di}} + \delta}{\sum_d M_d + \delta H - 1} \times \frac{n^{(w)}(d, z_{di}^{(w)})_{-t_{di}} + n^{(w)}(d, z_{di}^{(w)}) + \alpha}{\sum_d (N_d + M_d) + \alpha K - 1} \quad (2)$$

And, $n(z, s, w)$ is the number of tokens of word w is assigned to topic z with user assigned to characteristic s , $n^{(w)}(d, z)$ is the number of word tokens in document d is assigned to topic z , and $n(s, u)$ is the number of occurrence of user u is assigned to characteristic s .

2.4 UC-TagLDA

User characteristics can also be combined with other topic models. We describe the combination of User characteristics with TagLDA in this section.

TagLDA is used to model social tagged data like del.icio.us. In such a social tagged system, different user may use different tags to tag a same content since they will concern different aspects of the content or they may use different words to describe the same content. In this way, we add the users characteristics to the generative process based on the following assumptions: when a tag is chosen for a topic by a user, it not only rests with that topic, but also rests with the user's characteristics. Meanwhile, different types of characteristics generate different users.

To model the tagged social documents, TagLDA considers the generative process of both words and tags. For document d , the i th word w_{di} is generated in the same way as in LDA, while topic assignment $z_{dj}^{(t)}$ for the j th tag is chosen from the document's topic distribution θ_d , and that tag t_{dj} is also generated from a topic-specific multinomial distribution $\rho_{z_{dj}}$. In UC-TagLDA, the difference is

tag t_{dj} is not only influenced by tag distribution $\rho_{z_{dj}}$, but also influenced by that user's characteristics assignment $s_{dj}^{(u)}$ which also influence the generation of user u_{dj} .

The graphical model is shown in Figure 3. And we also use Gibbs sampling for posterior inference.

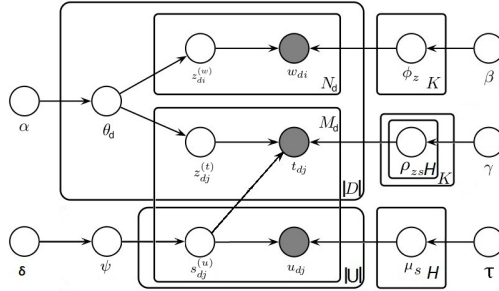


Fig. 3. Graph representation of UC-TagLDA

3 Experiments

3.1 Dataset

UC-LDA. We use two dataset of papers for experiments about UC-LDA, the abstract from CiteseerX and the full paper from NIPS. The CiteseerX dataset (1.2GB xml file) is downloaded from CiteSeerX OAI collection and it has 456,353 abstracts from 650,478 authors. The NIPS dataset (35MB mat file) [9] has 2,484 papers from 2,865 authors.

We randomly choose about 90% as the training data and about 10% as the testing data. For CiteseerX dataset, there are 407,812 training documents (598,745 authors) and 48,541 testing documents (60,1973 new authors). For NIPS dataset, there are 2,207 documents (2,290 authors) and 277 documents (575 new authors) left for testing.

UC-TagLDA. We use the data from del.icio.us provided by DAI Labor[10] for UC-TagLDA. Del.icio.us is a social bookmarking system in which users can tag each of their bookmarks freely. Each record of the data consists of three parts, including user, url and tags.

We chose the bookmarks collected in 2004 for the experiments. To avoid data sparseness, we have removed the URLs that have been tagged by fewer than 20 users, and also removed those users who tagged fewer than 50 URLs. After preprocessing, 1121 users and 2476 URLs remained. We then crawled the web pages of these URLs. After preprocessing to remove the irrelevant content and the web page stop words, there were about 143 words left for each page.

As the dataset used for UC-TagLDA is small, we ran the experiments using this dataset with 10-crossfold validations.

3.2 Perplexity

UC-LDA. In the comparison experiments for UC-LDA, we compute the perplexity of words comparing among LDA, ATM, AITM and UC-LDA on 10, 20, 50, 100, 150, 200, 500 and 1000 topics. where 1000 topics only set for NIPS dataset. As CiteseerX dataset is much larger than NIPS dataset and the limitation of memory, it is hard to do experiments with exactly the same number of topics on both dataset. For AITM, we also set different interest number (1, 5, 10, 20, 50, 100, 150, 200), and choose the best result (lowest perplexity) on each topic number for comparative purposes.

The experimental results of NIPS and CiteseerX are respectively shown in Table 2 and Table 3.

Table 2. UC-LDA:Perplexity on NIPS Dataset

Topic Num	LDA	ATM	AITM	UC-LDA				
				1 cha.	5 cha.	10 cha.	20 cha.	50 cha.
10	1985.94	4079.68	3242.43	1992.67	1774.21	1576.88	1428.29	1639.83
20	1770.34	4271.98	3242.24	1765.18	1714.58	1597.98	1415.06	1864.60
50	1516.10	4509.35	3242.34	1532.38	1494.18	1339.65	1879.24	2351.45
100	1370.35	4526.52	2991.91	1381.87	1334.28	1284.41	1573.89	2011.06
150	1304.43	4588.58	3107.88	1318.61	1205.48	1666.07	2147.20	2613.61
200	1349.78	4540.07	3242.56	1277.93	1636.87	1926.90	2256.08	2690.94
500	1337.54	4566.81	3242.55	1338.06	1955.98	2319.86	2962.63	3163.50
1000	1447.76	4597.52	3242.94	1438.19	2482.52	2942.73	3599.60	4577.21

Table 2 shows our model outperforms the other models on 10, 20, 50,100 and 150 topics. And our model (with 1 characteristics) and LDA has nearly the same perplexity on 200, 500, 1000 topics,

When the topic number is set to a small value, user characteristics can separate a topic into several sub-topics by considering user difference, and cause better performance. As the increasing of number of topic, LDA gets its best performance, but the separation of topics in LDA and UC-LDA are not same. The best perplexity of UC-LDA is got at 5 characteristics and 150 topics, while LDA does not get best perplexity at $5 \times 150 = 750$ topics. The best performance of UC-LDA improves about 9.9% comparing with best LDA.

And this results can also show that LDA is a special case of UC-LDA with 1 characteristic.

From Table 3, we can get the same conclusion as in Table 2. The best performance of UC-LDA improves about 12.2% comparing with best LDA. UC-LDA brings bigger improvement of perplexity on bigger data.

UC-TagLDA. In UC-TagLDA, we have a latent variable for the user's characteristics in addition to the latent variable for topics common in both of our

Table 3. UC-LDA:Perplexity on CiteseerX Dataset

Topic Num	LDA	ATM	AITM	UC-LDA				
				1 cha.	5 cha.	10 cha.	20 cha.	50 cha.
10	1638.68	2453.72	2321.70	1659.38	1495.68	1320.19	1196.02	1118.37
20	1572.30	2758.42	1984.76	1603.35	1528.15	1457.65	1200.29	1211.38
50	1328.74	2786.09	2089.28	1322.25	1329.15	1248.31	1370.23	1545.94
100	1280.06	2664.23	1972.95	1281.47	1204.51	1452.76	1621.43	1892.18
150	1274.46	2772.25	2106.21	1270.66	1264.15	1729.40	1925.38	2229.91
200	1306.67	2836.97	2101.23	1287.89	1485.95	1875.92	2381.29	3095.13
750	1315.41	-	-	-	-	-	-	-

model and TagLDA. For topics, we tested six different values. For user’s characteristics, we tested three different values. The experimental results are shown in Table 4 and Table 5.

Table 4 shows our model has smaller perplexities than those in TagLDA on 5 and 10 characteristics with different topics number. The best perplexity of UC-TagLDA improves about 1.4% comparing with best TagLDA. Perplexities on tags shown in Table 5 bring us to the same conclusion. Our model outperforms TagLDA and the best perplexity of our model improves about 18.8% comparing with best TagLDA.

Table 4. UC-TagLDA:Perplexity of words on del.icio.us

Topic Num	Tag-LDA	UC-TagLDA		
		5 cha.	10 cha.	20 cha.
10	3896.92	3805.34	3930.83	3898.23
20	3285.39	3019.31	3293.35	3283.66
30	2895.01	2812.14	2838.54	2852.27
50	2874.11	2811.62	2801.13	2847.21
100	2775.25	2742.02	2740.11	2736.03
200	2798.67	2751.30	2739.42	2784.93

Unlike TagLDA, ATM models authors and documents at same time. We therefore compare our model to ATM.

To fit the data requirements of ATM, we mix the tags and users of each URL together and assume that all URL tags were co-created by all users who tagged the URL. The same parameters are set as the previous experiments.

Experimental results are shown in Table 5. It shows that UC-TagLDA significantly outperforms the ATM model on social tagged data.

3.3 Word Distributions over Different Characteristics

The user’s characteristics can perform in different forms. Some characteristics may represent different aspects of a topic, and some characteristics may represent

Table 5. UC-TagLDA:Perplexity of tags on del.icio.us

Topic Num	Tag-LDA	UC-TagLDA		ATM
		5 cha.	10 cha.	
10	165.23	144.87	158.24	1008.43
20	113.24	97.95	116.26	613.34
30	95.6	87.61	92.88	895.54
50	117.4	83.25	94.39	1032.57
100	92.96	70.62	86.52	1175.91
200	87.02	71.23	85.29	1264.21

different wording preferences. Table 6 and Table 7 respectively show the word/-tag rank of UC-LDA (on Citeseerx dataset) and UC-TagLDA.

In Table 6, for each topic, we list the word rank (Top-10) of two different characteristics. And we can find that although the top-10 words are nearly in the same set, they are in different order, which demonstrates different wording preferences. For example, in topic-7 (CiteSeerX), top 9 words are the same but in different position in charac-0 and charac-4. "medical" is the top one word in charac-0, and the second word in charac-4, "patients" is the second word in charac-0, and the 7th word in charac-4. "neural" is the second word in charac-0, and the 5th word in charac-4. Obviously, these different topics caused by user wording preference can not be found by LDA models. And we can also find the different perspective of the same topic. For example, topic-4 (NIPS) is about physiological, where charac-9 is interested in the physiological property, while charac-15 concerns more technical details.

In Table 7, topic-29 has to do with the web service, but charac-2 users are concerned with the usage of blog, while the charac-5 users may be paying more attention to the search technology. For topic-47, charac-0 and charac-3 are both interested in the hardware product, because they have used almost the same most used words, but in different order when they talk about a same topic, demonstrating different wording preferences.

3.4 Application on Recommendation(UC-TagLDA Only)

Both UC-TagLDA and TagLDA can be used for tag recommendation. As [5] shows TagLDA has better performance compared to K-means on tag recommendation, we designed two groups of experiments to compare their recommendation performance.

In the experiments, the topic number is set to 50 for both TagLDA and UC-TagLDA, for UC-TagLDA, the number of user's characteristics is also set to two different values, 5 and 30.

For each webpage, we chose top-N tags as the recommended tags and compared them to the user's tags.

To evaluate the model, we randomly selected 90% of 2476 URLs as training data, and the remaining was used as test data.

Table 6. UC-LDA: The Top-10 words of each characteristics for the same topic (50 topics, 20 characs)

Topic-7(CiteseerX)		Topic-9(CiteseerX)	
charac-0	charac-4	charac-0	charac-6
medical	brain	image	image
patients	medical	images	images
neural	clinical	visual	objects
clinical	activity	objects	video
brain	neural	motion	visual
patient	diagnosis	video	object
diagnosis	patients	objects	motion
activity	patient	spatial	spatial
treatment	treatment	robot	shape
cortex	disease	tracking	robot
Topic-4 (NIPS)		Topic-19 (NIPS)	
charac-9	charac-15	charac-6	charac-19
factorizations	volatile	table	contents
earliness	workshop	contents	table
rheological	detect	list	tables
forthcoming	division	figure	ftp
nanomaterials	renovation	tables	list
mell	eaor	preface	introduction
lipoproteins	electromyogram	introduction	figures
offending	mqp	postscript	esi
locationaware	neurotransmitter	ftp	acknowledgements
incidences	closures	acknowledgements	preface

Table 7. UC-TagLDA: The Top-10 tags of each characteristics for the same topic on del.icio.us(50 topic, 30 characs)

Topic-29		Topic-47	
charac-2	charac-5	charac-0	charac-3
wisdom	copyright	hardware	shopping
semantic	seo	shopping	hardware
history	searchengine	tech	gadgets
tricks	hardware	diy	tech
wordpress	ajax	geek	hacks
article	info	cool	diy
random	amusements	howto	geek
webdizajn	im	gadgets	gadget
java	commerce	video	shop
proxy	html	technology	technology

Table 8 gives the results of precision, recall and F1 score on our preprocessed data from DAI-Labor dataset.

Table 8. The evaluation result of recommendation

Topic Num	Method	Recall(%)	Precision(%)	F1 score
2	UC-TagLDA(5 cha.)	10.44	11.01	0.1072
	UC-TagLDA(30 cha.)	10.35	10.92	0.1063
	TagLDA	9.32	9.83	0.0957
5	UC-TagLDA(5 cha.)	16.20	6.83	0.0961
	UC-TagLDA(30 cha.)	18.12	7.64	0.1075
	TagLDA	15.65	6.59	0.0928
10	UC-TagLDA(5 cha.)	23.52	4.96	0.0819
	UC-TagLDA(30 cha.)	24.86	5.24	0.0866
	TagLDA	21.42	4.52	0.0746
20	UC-TagLDA(5 cha.)	31.92	3.36	0.0608
	UC-TagLDA(30 cha.)	32.77	3.55	0.0641
	TagLDA	30.35	3.19	0.0578

Table 9. The evaluation result of recommendation with different amounts of training data

Prop.	Method	Recall(%)	Precision(%)	F1 score
1/8	UC-TagLDA(5 cha.)	24.17	5.09	0.0841
	UC-TagLDA(30 cha.)	24.98	5.26	0.0870
	TagLDA	18.32	4.41	0.0711
1/4	UC-TagLDA(5 cha.)	24.42	5.22	0.0862
	UC-TagLDA(30 cha.)	24.93	5.25	0.0868
	TagLDA	21.09	4.45	0.0735
1/2	UC-TagLDA(5 cha.)	23.99	5.06	0.0835
	UC-TagLDA(30 cha.)	23.88	5.04	0.0832
	TagLDA	21.06	4.44	0.0734
1	UC-TagLDA(5 cha.)	23.52	4.96	0.0819
	UC-TagLDA(30 cha.)	24.86	5.24	0.0866
	TagLDA	21.43	4.52	0.0746

Table 10. Ranking of user’s tags

Model	UC-TagLDA		TagLDA
	5 cha.	30 cha.	
Ave. rank	181.37	157.25	212.91

Since we have more parameters in UC-TagLDA than in TagLDA, we wonder if the scale of training data will bring different effects. To evaluate the influences on the amount of training data, we randomly chose 1/2, 1/4 and 1/8 of the whole training data to train the model, and evaluated on the same set of testing data. The experimental data is shown in Table 9. In the experiment, the topic number was also set to be 50, and the evaluation is on the top 10 tags.

Based on the above experiments, we ranked the tags of each user on the web page for each model, and calculated the average ranks of the user’s real tags in the models. The ranking results are shown in Table 10, and from the results, it is easily to see that our model significantly outperforms TagLDA.

4 Conclusions

This paper proposes a new idea of topic model to address the problem of user characteristics. User’s relevant words are assumed to be not only generated from latent topics as in a normal topic model, but also influenced by the user’s characteristics. Experimental results show that the model with user’s characteristics(UC-LDA & UC-TagLDA) outperforms the previous models(LDA & TagLDA) and other similar topic models(ATM & AITM) on text modeling.

Furthermore, our model can also give some interesting results about user characteristics. We have found two varieties of user characteristics from our model, one concerns different views of a topic, and the other is different wording preference.

Acknowledgements. The work presented in this paper is supported by the National Science Foundation of china (No. 61273365) and National High Technology Research and Development Program of China (No. 2012AA011104).

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent dirichlet allocation. *Journal of Machine Learning Research* 3 (2003)
2. Blei, D.M., McAuliffe, J.D.: Supervised topic models. In: *NIPS (2007)*
3. Ramage, D., Hall, D., Nallapati, R., Manning, C.D.: Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (2009)*

4. Si, X., Sun, M.: Tag-lda for scalable and realtime tag recommendation. *Journal of Computational Information Systems* (2009)
5. Ramage, D., Heymann, P., Manning, C.D., Garcia-Molina, H.: Clustering the tagged web. In: *Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM 2009*, pp. 54–63. ACM, New York (2009)
6. Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P.: The author-topic model for authors and documents. In: *20th Conference on Uncertainty in Artificial Intelligence* (2004)
7. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 306–315 (2004)
8. Kawamae, N.: Author interest topic model. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010*, pp. 887–888. ACM, New York (2010)
9. Globerson, A., Chechik, G., Pereira, F., Tishby, N.: Euclidean embedding of co-occurrence data. *The Journal of Machine Learning Research* 8, 2265–2295 (2007)
10. Wetzker, R., Zimmermann, C., Bauckhage, C.: Analyzing social bookmarking systems: a delicio.us cookbook. In: *European Conference on Artificial Intelligence, ECAI* (2008)
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. Blei, D.M., Lafferty, J.D.: Correlated topic models. In: *NIPS*, p. 147 (2006)

Mining User Preferences for Recommendation: A Competition Perspective

Shaowei Jiang, Xiaojie Wang, Caixia Yuan, and Wenfeng Li

Center for Intelligence Science and Technology,
Beijing University of Posts and Telecommunications, Beijing, China
{shaowei.jiang,lwfeng115}@gmail.com, {xjwang,yuancx}@bupt.edu.cn

Abstract. Mining user preferences plays an important role in building personalized recommender systems. Instead of mining user preferences with the item content or the user-item-rating matrix, we exploit Bradley-Terry model to mine user preferences as pairwise comparisons. In this paper we assume that the user preference on each item can be represented by the combination of different content features, which brings a direct bridge between features and user preferences. Experimental results show that the method based on pairwise comparisons outperforms baseline approaches with less recommendation time.

Keywords: user preference, pairwise comparison, Bradley-Terry model, recommender system.

1 Introduction

Recent years have witnessed the unprecedented prevalence and significance of building recommender systems, which aim to recommend proper items for users with respect to their personal preferences. On the Internet, there are a large number of items with ratings of particular users such as books, movies and food. The ratings could effectively reflect user preferences on the items, and become important resources to mine user preferences and implement recommendations.

Collaborative filtering (CF) and content-based recommendation (CBR) are two commonly used approaches for recommendation. CF analyzes the similarity of users' preferences or items and recommends items for active users, using data rated by a great quantity of users. However, since it does not directly analyze the content of items, it may suffer from the so called cold start problem, which makes it fail to recommend items that have not been previously rated in the community [1]. Meanwhile, the CBR approach focuses on analyzing the similarity between item contents and user preferences, so it can process new items without user ratings. Traditional CBR method is generally divided into three steps: Firstly, it confirms item representation through extracting content features. Secondly, it mines the user preference and represents the preference with features by leveraging the past rating information. Finally, it recommends the most similar item set to the user, by comparing the user preference representation and the representation of the new item [2]. However, the CBR approaches

always excessively rely on the results of content analysis, which cannot take full advantage of the explicit rating data.

In this paper, we propose a novel approach for mining user preferences by exploiting the competition perspective from game theory, which sufficiently makes use of both the content of items and user ratings. Unlike previous works, which each level rating is considered as independent category, this paper discusses to mine user preferences with competition relationship of items rated different ratings. It is assumed that each item consists of several content features. Ratings on items are used to estimate a unique user preference value for each feature by Bradley-Terry model. The rating for a new item can therefore be predicted by user preference values of all features in the item. The basic idea is: a user U gives a higher rating to item A than that of B for the reason that A is better than B in the competition of U 's preferences. If A and B are represented by textual features, then it is supposed that features of A have overall stronger competitiveness than those of B . Therefore, the competition between two items with various ratings is decomposed into competitions among content features of two items. The user preference value of each feature could be estimated, which is obtained by competitions for several times. Thus, when a new text C turns up, the user preference value for C can be calculated through summing up the value of each feature in C . In this way, this model can integrate both user ratings and the content of items for mining user preferences.

The remainder of this paper is organized as follows. In Section 2 we review related work. The user preferences mining approach based on pairwise comparisons is presented in Section 3. Section 4 reports experiments and results. Finally, Section 5 summarizes our conclusions and discusses directions of future work.

2 Related Work

Generally, the related works of this paper can be grouped into three categories. The first is the CBR. CBR algorithms usually apply vector space model (VSM) to represent items. Recommendation results are normally obtained by machine learning algorithms, such as the nearest neighbor method (find K rated items most similar with the new item, use the user preference of K items to judge the preference of the new item), Rocchio algorithm (obtain user preference vector to features from liked and disliked items, calculate the similarity of the vector and the feature value vector of the new item as user preference of the new item), the decision tree algorithm (build the tree structure to use its branch to classify the new item with features), the linear classifier algorithm (find a plane in the higher dimensional space to separate class points) and the Naive Bayes algorithm (calculate probabilities of features in all categories to predict the new item). But these methods are too dependent to the content, with rating information as an auxiliary tool.

The second is the CF recommendation. CF algorithms can be divided into memory-based and model-based algorithms. The user-based and item-based algorithms are common memory-based methods. The user-based algorithm calculates the similarity between the active user and other users, then recommends

favorite items of users who are very similar to the active user [3]. The item-based algorithm analyzes items rated by the active user, and recommends items that are similar to ones that he rated [4]. In order to avoid overfitting phenomenon from the item-based algorithm, Lemire and Maclachlan [5] proposed a slope one algorithm. In terms of model-based algorithms, to reduce the dimension of the user-item-rating matrix, latent semantic models such as probabilistic latent semantic analysis, Latent Dirichlet allocation, the Single Value Decomposition (SVD) algorithm and its follow-up improvement algorithms appeared [6-11]. Yang et al [12] studied user choice behavior in a series of items under the background of CF, obtaining better recommendation effect. No matter what kind of CF algorithms, there is always the cold start problem.

The third is hybrid recommendation. It could combine CBR and CF recommendation results, and make feature augmentation, such as using content features to offset users' simple ratings, or training users' ages and genres of movies in a classifier [13-16], and so on. Although there are many combination methods, they are not effective in a specific issue all the time.

This article proposes a new way to estimate preference values of micro content features from ratings data of items, basing on a pairwise comparison model. Preference values of features are then used to rate new items.

3 The User Preferences Mining Approach Based on Pairwise Comparisons for Recommendation

3.1 Motivation Discussion

Given a user u and an item set $\bar{I} = \{1, 2, \dots, I\}$, $r(u, i)$ denotes the rating for an item $i \in \bar{I}$ by u . To solve the problem of rating a new item $i_a \notin \bar{I}$, the user preference value of u on i is introduced, denoted by $p(u, i)$. For any two items i_A and i_B , if $r(u, i_A)$ is similar with $r(u, i_B)$, then $p(u, i_A)$ is assumed to be similar with $p(u, i_B)$. The higher the rating is, the stronger the preference is, and vice versa. For example, items with rating 5 are supposed to have similar preferences, and the preference for the item rated with 5 is stronger than the one for the item rated with 4.

Now we have $p(u, i)$ as the preference value of a user u on an item i . Considering an item consists of some content features, it is therefore assumed that preference value of u on i correspondingly consists of preference values of u on features in i . Given a feature set $\bar{F} = \{1, 2, \dots, F\}$ for all items, if i consists of feature subset $F_i \subset \bar{F}$ and each feature $f_i \in F_i$ is independent, $p(u, i)$ is determined by combination of $\{p(u, f_i)\}$. So, for a new item i_a , its preference value can be calculated by $\{p(u, f_{ia})\}$.

There are three stages of our model. First, $p(u, f)$, $f \in \bar{F}$ for all features are estimated. Second, $p(u, i)$ of an item i is calculated. Finally, $r(u, i)$ is acquired basing on $p(u, i)$. The details are given in 3.2, where Bradley-Terry model is used for $p(u, f)$ estimation and $p(u, i)$ calculation. The nearest neighbor method (KNN) is used to get $r(u, i)$ from $p(u, i)$.

3.2 Model Description

Bradley-Terry model is first given for $\mathbf{p}(\mathbf{u}, \mathbf{f})$ estimation and $\mathbf{p}(\mathbf{u}, \mathbf{i})$ calculation, and then KNN is used for getting $\mathbf{r}(\mathbf{u}, \mathbf{i})$ from $\mathbf{p}(\mathbf{u}, \mathbf{i})$.

Out of personal interest, a user has various ratings to different items. In turn, items have matches on the level of user interest, resulting in the survival of the fittest. There is obvious competitive relationship among them. For the comparison between things, the Bradley-Terry (BT) model [17] is a popular competitive relationship probability model. This model is used to measure the ability of competitive object in the pairwise comparison. It assumes that the contestant's win rate is proportional to his own competitiveness. And it trains and quantifies the capacity of the contestant basing on the assumption. The competitor's winning probability is described by the original BT model in the pairwise comparison. It is shown in Formula (1).

$$P(o \text{ beats } q) = \frac{\gamma_o}{\gamma_o + \gamma_q}, \tag{1}$$

where γ_o refers to overall competition level of the individual o . Different from an individual, the content of a item consists of multiple features. So this article makes use of the generalized BT model for multi-player team competition. It is illustrated by Formula (2).

$$P(1\text{-}2\text{-}3 \text{ wins against } 4\text{-}5 \text{ and } 2\text{-}6\text{-}7) = \frac{\gamma_1\gamma_2\gamma_3}{\gamma_1\gamma_2\gamma_3 + \gamma_4\gamma_5 + \gamma_2\gamma_6\gamma_7}. \tag{2}$$

Note one feature can appear in multiple items here. Through competitions among items for many times, the competitiveness of each feature may be ultimately determined, that is, the user preference value.

As there are all competitive relations among items got different ratings, any two items with various ratings can play a game. The content describing the item can be analogous to a team participating in a competition, and features acted like players. The content of the item that is higher rated by the user is the winning team in the game. During the competitions, the user preference values of features in the winning team increase, meanwhile the values of features in the losing team correspondingly reduce. In the end, through iterative calculation of all sessions, user preference values of features settle out.

The estimation process of $\{\mathbf{p}(\mathbf{u}, \mathbf{f})\}$ is described as follows. First of all, we denote user preference values of n features with parameters $\mathbf{p}(\mathbf{u}, \mathbf{f}_1), \mathbf{p}(\mathbf{u}, \mathbf{f}_2), \dots, \mathbf{p}(\mathbf{u}, \mathbf{f}_n)$. $\mathbf{p}(\mathbf{u}, \mathbf{i}_1), \mathbf{p}(\mathbf{u}, \mathbf{i}_2), \dots, \mathbf{p}(\mathbf{u}, \mathbf{i}_K)$ are user preference values of K items in a competition. If an item \mathbf{i} consists of M features, $\mathbf{p}(\mathbf{u}, \mathbf{i})$ could be computed by Formula (3).

$$p(u, i) = \prod_{m=1}^M p(u, f_m). \tag{3}$$

In other words, the preference value of each item is the product of features' values, following the assumption of the generalized BT model [18]. The N independent match results among contents, $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N$, are get from the corpus. So the probability of the game j result can be written as Formula (4).

$$P(R_j) = \frac{p(u, i_{win})}{\sum_{k=1}^K p(u, i_k)}. \quad (4)$$

The objective of the model is to maximize the probability of these games' results, and the objective function L is shown by Formula (5).

$$L = \prod_{j=1}^N P(R_j). \quad (5)$$

$p(\mathbf{u}, \mathbf{f}_w)$ could be iteratively calculated, by setting the feature w as the only variable in one iteration. Thus Formula (4) could be rewritten as Formula (6).

$$P(R_j) = \frac{f(p(u, f_w))}{g(p(u, f_w))}. \quad (6)$$

$f(p(\mathbf{u}, \mathbf{f}_w))$ and $g(p(\mathbf{u}, \mathbf{f}_w))$ have $p(\mathbf{u}, \mathbf{f}_w)$ as a variable and other $\{p(\mathbf{u}, \mathbf{f}_m)\}$ as constants. So the logarithm of L is shown in Formula (7).

$$\log L(p(u, f_w)) = \sum_{j=1}^N \log f(p(u, f_w)) - \sum_{j=1}^N \log g(p(u, f_w)). \quad (7)$$

Applying the Minorization-Maximization algorithm [19], we start to build the approximation function m at an initial point $p^0(\mathbf{u}, \mathbf{f})$ to make $m(p^0(\mathbf{u}, \mathbf{f})) = \log L(p^0(\mathbf{u}, \mathbf{f}))$ and $m(p(\mathbf{u}, \mathbf{f})) \leq \log L(p(\mathbf{u}, \mathbf{f}))$ for $\{p(\mathbf{u}, \mathbf{f})\}$. Then the next point to be computed is found, at which the approximation function value is max. As $p(\mathbf{u}, \mathbf{f}_w)$ either occurs in winners or losers, we set $p(\mathbf{u}, \mathbf{f}_w)$ as x , and omit constants to get the minorizing function (8)

$$m(x) = WN_w \log x - \sum_{j=1}^N \frac{g'(x)x}{g(x)}, \quad (8)$$

WN_w is the sum of occurrence number of w in all winning teams. $g(x)$ is the total preference value of all teams in the game. With the thought that maximizes $m(x)$, the iterative formula (9) of $p(\mathbf{u}, \mathbf{f}_w)$ is

$$p(u, f_w)^{(k+1)} \leftarrow \frac{WN_w}{\sum_{j=1}^N \frac{g'(p(u, f_w)^{(k)})}{g(p(u, f_w)^{(k)})}}. \quad (9)$$

All of the preference values are confirmed when L is maximized.

After the above steps, $\{p(\mathbf{u}, \mathbf{f})\}$ are known. The user preference value $p(\mathbf{u}, i_a)$ of the new item i_a is calculated by Formula (3). If $f_{ia} \notin \bar{F}$, $p(\mathbf{u}, f_{ia})$ is set as 1. According to our first assumption, $p(\mathbf{u}, i_a)$ can predict $r(\mathbf{u}, i_a)$, which could be achieved by KNN easily. It is to find k items $I_k \subset \bar{I}$ of which $\{p(\mathbf{u}, i_k), i_k \in I_k\}$ (also calculated by Formula (3)) are nearest to $p(\mathbf{u}, i_a)$. As $p(\mathbf{u}, i)$ is a scale value, the absolute value of difference between $p(\mathbf{u}, i_A)$ and $p(\mathbf{u}, i_B)$ is used as the similarity measure. So the similarity formula (10) is

$$Sim(i_A, i_B) = |p(u, i_A) - p(u, i_B)|. \quad (10)$$

Then $r(\mathbf{u}, i_a)$ is set as the most frequent $r(\mathbf{u}, i_k)$ in I_k .

4 Experiments

Firstly, prediction precision based on CBR and CF recommendation algorithms are compared with our method on two movie datasets. Secondly, the relationship between competitive scales and recommendation effect is discussed.

4.1 Recommendation Effect Experiments

The algorithms are evaluated on two datasets, MovieLens and Netflix [20-21], which are the most famous datasets in the field of recommender system. The MovieLens dataset is the real data crawled by the MovieLens movie recommender system (<http://movielens.umn.edu>). It contains rating data scored by numbers from 1 to 5 which contains 943 unique users' 10,000 ratings to 1,682 movies. The data of 600 users who rate movies more than 45 times are selected in the experiment. The content of each movie consists of director names, major movie star names and film styles (such as action, sci-fi, etc.), which are crawled from the IMDB website (<http://www.imdb.com>). The feature set includes all words in the content of items. The Netflix dataset is the Netflix Prize data, 1 to 5 rating data of about 10 billion times rating from 480,189 anonymous users on about 17,770 movies. About 100,000 users are randomly selected as experimental samples, with the same setting of movie content and source as the MovieLens dataset. On account of the large amount of data in the Netflix dataset, there are lots of internal competition relations. To train parameters faster, the training part of the competitive method is calculated by parallel processing in Beijing Computing Center (<http://www.bcc.ac.cn>).

Mean absolute error (MAE) and root mean squared error (RMSE) are used to evaluate the recommendation performance.

MAE. It calculates the average difference between prediction ratings and actual ratings in the test set.

$$|\bar{E}| = \frac{\sum_{i=1}^N |p_i - v_i|}{N}, \quad (11)$$

p_i is the prediction rating of a test sample, v_i is the actual rating, N is the number of test samples.

RMSE. It represents the deviation of the prediction rating than the actual rating, more emphasizing on large errors. The formula is as follows.

$$|\bar{E}| = \sqrt{\frac{\sum_{i=1}^N (p_i - v_i)^2}{N}}. \quad (12)$$

The baseline approaches are a CBR algorithm and RSVD in CF methods. The CBR recommendation algorithm uses term frequency-inverse document frequency (TF-IDF) values generated from VSM as content feature values, and then makes use of KNN to predict ratings. So it is named TFIDF-KNN here. It also includes all words in its feature set. Our method names BT-KNN. Three methods are evaluated on average errors of predict ratings by 5 cross-validation.

Table 1. Recommendation effect of methods on the MovieLens dataset

MAE	K=1	K=5	K=10	K=20	K=50	K=100	K=200	K=500
BT-KNN	0.819	0.821	0.822	0.822	0.822	0.819	0.819	0.819
TFIDF-KNN	0.978	0.920	0.887	0.866	0.852	0.848	0.848	0.848
RSVD	1.131							
RMSE	K=1	K=5	K=10	K=20	K=50	K=100	K=200	K=500
BT-KNN	1.110	1.110	1.109	1.108	1.106	1.105	1.105	1.105
TFIDF-KNN	1.316	1.251	1.217	1.192	1.176	1.172	1.171	1.171
RSVD	1.378							

Table 2. Recommendation effect of methods on the Netflix dataset

MAE	K=1	K=5	K=10	K=20	K=50	K=100	K=200	K=500
BT-KNN	0.755	0.758	0.761	0.764	0.769	0.769	0.769	0.770
TFIDF-KNN	0.911	0.848	0.817	0.795	0.776	0.769	0.766	0.766
RSVD	0.905							
RMSE	K=1	K=5	K=10	K=20	K=50	K=100	K=200	K=500
BT-KNN	1.040	1.040	1.042	1.045	1.048	1.049	1.049	1.050
TFIDF-KNN	1.251	1.182	1.148	1.124	1.103	1.095	1.092	1.091
RSVD	1.109							

Since BT-KNN uses independent personal data, the metrics are finally per-user average values. The experimental results are shown in Table 1 and Table 2.

As can be seen from the tables, the recommendation effect of BT-KNN is relatively stable, both on the MovieLens dataset and the Netflix dataset. It achieves good performance even at small K, and is better than TFIDF-KNN significantly at the same time. The predictive rating error of TFIDF-KNN gradually decreases along with the increase of the neighbor number, and stabilizes at a big K. When K = 1, BT-KNN improves the MAE performance of TFIDF-KNN by 19.4% on the MovieLens dataset and by 20.7% on the Netflix dataset. BT-KNN improves the RMSE performance of TFIDF-KNN by 18.6% on the MovieLens dataset and by 20.3% on the Netflix dataset. About the methods' best recommendation effect, comparing to TFIDF-KNN, MAE of BT-KNN is increased by 3.5% on the MovieLens dataset and by 1.5% on the Netflix dataset, and RMSE is increased by 6.0% on the MovieLens dataset and 4.9% on the Netflix dataset. Compared to RSVD, BT-KNN has increased MAE by 38.1% and RMSE by 24.7% on the MovieLens dataset, while MAE by 19.9% and RMSE by 6.6% on the Netflix dataset.

BT-KNN also has the advantage on the recommendation time. The comparisons on time are in Table 3.

Table 3. Average recommendation time of methods

Avg. Time (ms)	BT-KNN	TFIDF-KNN	RSVD
MovieLens	1.040	7.739	0.141
Netflix	3.578	111.525	64.397

From the experimental results, the pairwise comparisons based method has higher predictive accuracy and stronger robustness. It reduces prediction impacts of the case brought by CBR, which items' content vectors are similar with inconsistent user preferences. CF focuses on decreasing the overall rating matrix error, so the adaptability of predicting personalized ratings is not strong. And competitive approach is faster than others in recommending.

4.2 The Relationship between Competitive Scales and Recommendation Effect

The quantity of ratings reflects the competitive scale. The fewer ratings are, the lower competition scale is. For example, if the rating scale is only two indicated by 'good' and 'bad', there is only one type of competitive relationship. For 5 ratings scale, every two level rating items have competitive relationships to each other.

To understand the characteristics of the user preferences mining approach based on pairwise comparisons, the recommendation effect of the method is researched under different competitive scales. Using the MovieLens dataset, we simulated 2 ratings data with $\{\{1, 2, 3\}, \{4, 5\}\}$ (original ratings), as well as 3 ratings data with $\{\{1, 2\}, \{3\}, \{4, 5\}\}$. Combined with the previous 5 ratings data, there are 3 different kinds of competitive scale data. Improved rates of BT-KNN with respect to TFIDF-KNN are observed in 3 cases. The relationship between competitive scale and recommendation effect is discovered. Recommendation effect of BT-KNN and TFIDF-KNN are demonstrated in Fig. 1 and Fig. 2, using 2 ratings data and 3 ratings data.

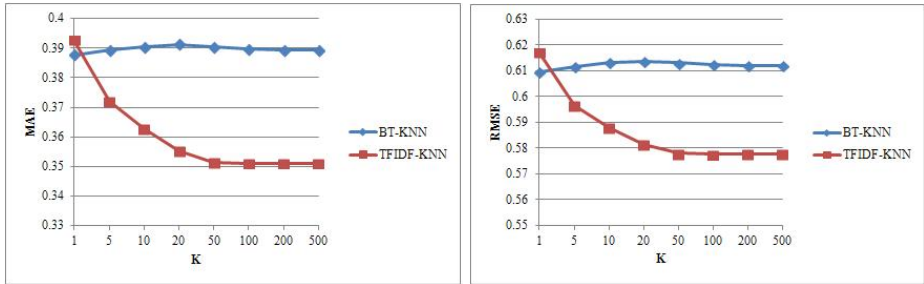


Fig. 1. The comparison of BT-KNN and TFIDF-KNN on the 2 ratings dataset

According to the above experimental results, recommendation effect relative variations of BT-KNN can be obtained in 3 kinds of competitive scale data. Fig. 3 shows them as follows.

It could be seen that recommendation effect of BT-KNN is worse than TFIDF-KNN on 2 ratings data under the condition of a small competitive scale. But as the increasing of the competitive scale, effect of BT-KNN is better than TFIDF-KNN on both 3 and 5 ratings data, and it is more obvious on 5 ratings data.

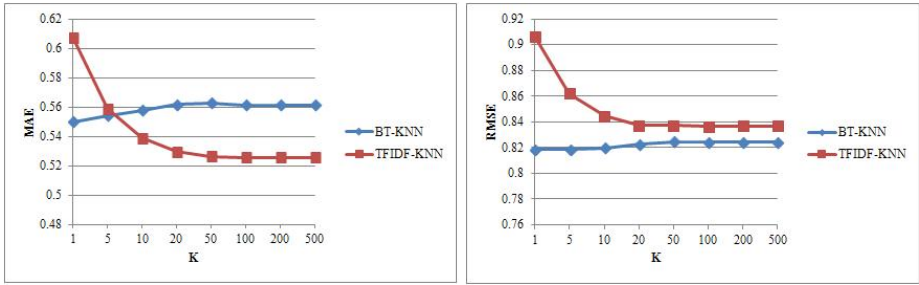


Fig. 2. The comparison of BT-KNN and TFIDF-KNN on the 3 ratings dataset

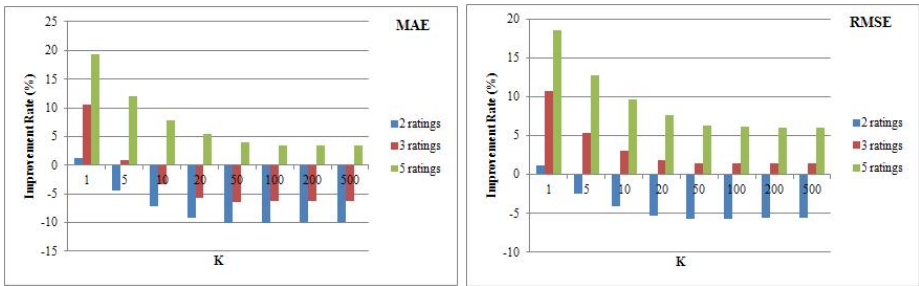


Fig. 3. The improved rates of BT-KNN contrasting with TFIDF-KNN on metrics

The result illustrates the method’s recommendation effect is better on the larger competition scale.

5 Conclusions

This article describes the basic idea of the user preferences mining approach based on pairwise comparisons. It unites the content and user rating of an item together with the competitive relationship, and obtains a unique identification of the user preference on a feature. The method provides better recommendation accuracy and less recommended time. And it demonstrates that the larger competitive scales of data, the better recommendation effect.

Currently, when the same feature is in different items, each feature is assumed to have just a user preference value, regardless of the influence of its context on its user preference. In the future, to use the binary features with the context will be investigated to verify the reasonableness of the assumption. And the overall value is the product of its members’ values, which is the hypothesis of the generalized BT model, is rather simple. Hereafter, the composite mode adapted data will be studied, to improve the recommendation performance of the approach.

Acknowledgments. The work in this paper is supported by the National Science Foundation of China (No. 61273365) and National High Technology Research and Development Program of China (No. 2012AA011103).

References

1. Cacheda, F., Carneiro, V., Fernndez, D., Formoso, V.: Comparison of collaborative Filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on The Web (TWEB)* 5(1), 1–33 (2011)
2. Lops, P., Gemmis, M.D., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Rokach, L., Kantor, P.B. (eds.) *Recommender Systems Handbook*, pp. 73–105. Springer, New York (2011)
3. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: *The 1994 ACM Conference on Computer Supported Cooperative Work*, pp. 175–186. ACM Press, New York (1994)
4. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: *Proc. of the WWW 2001*, pp. 285–295. ACM Press, New York (2001)
5. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: *SIAM Data Mining Conference, SDM 2005* (2005)
6. Hofmann, T.: Probabilistic Latent Semantic Analysis. In: *Proc. of 15th Conf. Uncertainty in Artificial Intelligence*, pp. 289–296 (1999)
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3, 993–1022 (2003)
8. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Application of dimensionality reduction in recommender systems—a case study. In: *The ACM WebKDD Workshop* (2000)
9. Funk, S.: Netflix update: Try this at home, <http://sifter.org/~simon/journal/20061211.html>
10. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: *Proc. of the KDD 2008*, pp. 426–434. ACM Press, New York (2008)
11. Koren, Y.: Collaborative Filtering with Temporal Dynamics. In: *Proc. of the KDD 2009*, pp. 447–456. ACM Press, New York (2009)
12. Yang, S.H., Long, B., Smola, A.J., Zha, H., Zheng, Z.: Collaborative competitive filtering: learning recommender using context of user choice. In: *Proc. of the SIGIR 2011*, pp. 295–304. ACM Press, New York (2011)
13. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: *The ACM SIGIR 1999 Workshop Recommender Systems: Algorithms and Evaluation*. ACM Press, New York (1999)
14. Pazzani, M.J.: A framework for collaborative, content-based, and demographic filtering. *Artificial Intelligence Review* 13(5-6), 393–408 (1999)
15. Melville, P., Mooney, R.J., Nagarajan, R.: Content-Boosted collaborative filtering for improved recommendations. In: *Proc. of the AAAI 2002*, pp. 187–192 (2002)

16. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: Using social and content-based information in recommendation. In: The AAAI 1998, pp. 714–720 (1998)
17. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 39, 324–345 (1952)
18. Coulom, R.: Computing Elo Ratings of Move Patterns in the Game of Go. In: Herik, H.J.V.D., Uiterwijk, J.W.H.M., Winands, M.H.M., Schadd, M.P.D. (eds.) *Computers Games Workshop 2007*, pp. 113–124 (2007)
19. Hunter, D.R.: MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics* 32(1), 384–406 (2004)
20. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: *Proc. of the SIGIR 1999*, pp. 230–237. ACM Press, New York (1999)
21. Bennett, J., Lanning, S.: The netflix prize. In: *KDD Cup and Workshop (2007)*

A Classification-Based Approach for Implicit Feature Identification

Lingwei Zeng and Fang Li

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China
{chasuner, fli}@sjtu.edu.cn

Abstract. In recent years, sentiment analysis and opinion mining has grown to be one of the most active research areas. Most of the existing researches on feature-level opinion mining are dedicated to extract explicitly appeared features and opinion words. However, among the numerous kinds of reviews on the web, there are a significant number of reviews that contain only opinion words which imply some product features. The identification of such implicit features is still one of the most challenge tasks in opinion mining. In this paper, we propose a classification-based approach to deal with the task of implicit feature identification. Firstly, by exploiting the word segmentation, part-of-speech (POS) tagging and dependency parsing, a rule based method to extract the explicit feature-opinion pairs is presented. Secondly, the feature-opinion pairs for each opinion word are clustered and the training documents for each clustered feature-opinion pair are then constructed. Finally, the identification of implicit features is formulated into a classification-based feature selection. Experiments demonstrate that our approach outperforms the existing methods significantly.

Keywords: Opinion Mining, Implicit Feature, Feature Extraction.

1 Introduction

Nowadays, an increasing number of people are buying products on the web. Product reviews posted on the web can provide a lot of valuable information for those potential customers. At the same time, the product manufacturers can also obtain useful feedbacks about products from users. However, as the number of reviews that a product receives grows rapidly, sometimes the amount of comments of a product may exceed one thousand or even more, which makes it very hard for a potential customer to read all these reviews to obtain useful information. In order to automatically process and analyze reviews on the web, a lot of research efforts [4,9,5] have been done on opinion mining and sentiment analysis from the magnitude of reviews.

Document-level sentiment analysis [12] mainly classifies the whole review's emotional orientation. Sentence-level sentiment analysis determines whether each sentence expressed a positive, negative, or neutral opinion, which is closely

related to subjectivity classification[19,21,20] that distinguish subjective sentences with opinions from objective sentences. However, both document-level and sentence-level analysis can not discover what exactly people like or do not like, thus simply judging the sentiment orientation of a review unit fails to detect many significant details. In order to extract specific features and opinions from reviews, many researchers began to study the problem of finer-grained opinion mining, which is known as feature-level opinion mining[13,4,9,5].

Example 1. 很漂亮, 功能很多值得购买, 价格有点贵, 很喜欢白色, 送货比较快! (Very beautiful, there are plenty of functions that worth to buy, the price is a little expensive, really like the white color, the delivery is also very fast!)

When people read a customer review, they mostly concern the opinion word and its corresponding aspect or feature. In product review mining, feature is usually the component or attribute of the product. Example 1 is a digital camera review about *Nikon D90*. Explicit features such as “功能” (function), “价格” (price) and “送货” (delivery) can be extracted from the above comment. Except for explicit feature, there is another significant kind of feature that doesn’t directly appear in the review sentences but can be deduced from the opinion word, which is known as implicit feature. In the above example review, the opinion word “漂亮” (beautiful) has implied that the feature which the user talked about is the camera’s “外观” (exterior) although this feature doesn’t explicitly appear in the sentence.

In feature-specific opinion mining, most of the existing researches[1,24,7,23] mainly focused on the problem of extracting product features and opinions that explicitly appeared. However, according to the observation, in our crawled Chinese reviews of five kinds of digit cameras, we statistically discover that at least 28 percent of the sentences are implicit sentences that imply implicit features, which is a considerable proportion.

In this paper, we mainly focus on the problem of implicit feature identification. Our approach is very different from existing research works. The approach that former researches used is based on association rule mining. The core idea of this method is to use mined association rules to identify the implicit feature by finding the mapping of a specific feature for the opinion word. Although the association rule based approach is very useful and effective to identify the implicit feature for some kinds of opinion words that have relatively certain collocated features, for example, the opinion word “便宜” (cheap) is always used to describe the product’s feature “价格” (price). But it fails to deal with many complex situations, for example, the opinion word “好” (good) are often used to describe a lot of features, such as “信号” (signal), “屏幕” (screen) and “摄像头” (camera), by using the mined rules it can only map the opinion word “好” (good) to a specific feature, which is not correct for many other different situations. By considering both the associated relation and the context of the opinion word, our classification-based approach is able to identify different implicit feature for the opinion word with different situations.

The remaining parts of this paper are organized as follows. In section 2, we introduce some related works. The details of our proposed classification-based approach are introduced in section 3. In section 4, the experimental results are evaluated and discussed. our conclusion and future work are presented in section 5.

2 Related Work

Opinion mining has been extensively studied by many researchers in recent years. Most of these researches have focused on two main research directions: one is sentiment classification and the other direction is feature-based information extraction. Research efforts[10,17,11,14] on sentiment classification deal with the task of classifying each customer review as positive, negative or neutral. While feature-based opinion mining[4,6,5,8] focused on the task of extracting opinions consisting of information about features. In contrast to sentiment classification, opinion extraction aims at producing richer information and requires an in-depth analysis of reviews. The most representative researches in feature-level opinion mining are Hu and Liu's works[4,5,9]. The conception of implicit feature was first mentioned in their papers based on the analysis of English reviews. In contrast to explicit feature that directly appears in review sentences, implicit feature is the feature that does not occur in the comment, but can be deduced from opinion words and contexts based on the understanding of human language.

Semantic association analysis based on Point-wise Mutual Information (PMI) was used to infer the implicit features [15]. They predefined a domain-specific feature set as candidate implicit features, and then take the mutual information approach to map a opinion indicator to a certain feature of the feature set. In [16], a clustering method was proposed to map implicit opinion words to their corresponding explicit features. They clustered product features and opinion words simultaneously and iteratively by fusing both their content information and association relation, and then construct the sentiment association set between the groups of features and opinion words by identifying their strongest n sentiment links.

A co-occurrence association rule mining (coAR) approach was proposed to identify implicit features[3]. They firstly mined a set of association rules of the form [opinion-word, explicit-feature] from review sentences based on the co-occurrence of the opinion-word and explicit-feature. Then they cluster the explicit features to generate more robust rules. When given a new opinion word with no explicit feature, they searched a matched list of rules, among which the rule with the highest frequency weight is fired to map the opinion word to its identified implicit feature. In [18], they proposed a hybrid association rule mining approach for the task of implicit feature identification. Their approach used several complementary algorithms to mine as many association rules as possible. They firstly extract candidate feature indicators and then compute the co-occurrence degree between the candidate feature indicators and the feature words. Each indicator and the corresponding feature word constitute a rule(feature indicator \rightarrow feature word). They used such rules to identify implicit features.

3 Classification-Based Approach

In this section, we first illustrate the problem of implicit feature identification and present some definitions we have used in this paper. The framework of our proposed classification-based approach has also been presented. Then we explain the main steps of our approach in detail.

3.1 Problem Statement

In this paper, we focus on the feature-level opinion mining of product reviews. On the online shopping websites, such as Amazon or Taobao Marketplace, each product can receive a large number of customer reviews that have been posted by people who have bought this product. The set of products can be represented as $P = \{P_1, P_2, P_3, \dots, P_n\}$. For each product P_i , there is a set of customer reviews $R_i = \{r_1, r_2, r_3, \dots, r_m\}$. The customer reviews can be regarded as text documents, although some of them may be very short and consisted of just a few sentences, but there are also many long reviews that can be as long as articles. Each review r_j can be represented as a sequence of sentences $r_j = \{s_1, s_2, s_3, \dots, s_l\}$. Each sentence s_k may be consisted of several clauses $s_k = \{c_1, c_2, c_3, \dots, c_h\}$.

Definition 1. *implicit feature*

A product feature f is defined as the whole product, service or the attribute or component of the product. If a feature f appears in review sentences, then it is defined as *explicit feature*. If f does not appear in review sentences, but it is implied, which means that people who read the review can understand what feature has been talked about, then this feature f is regarded as *implicit feature*.

Definition 2. *implicit sentence*

Implicit sentence is a sentence in a review that contains at least one implicit feature. *Explicit sentence* is defined similarly, a sentence that contains at least one explicit feature is called *explicit sentence*. It should be noted that a implicit sentence can also be a explicit sentence.

Definition 3. *feature-opinion pair*

A feature-opinion pair is consisted of a feature and an opinion word, and the opinion word is used to modify the feature. If opinion word and its modified feature co-occur in a sentence, then such feature-opinion pair is defined as the sentence's *explicit feature-opinion pair*. The feature-opinion pair is denoted as $\langle \text{feature}, \text{opinion} \rangle$.

3.2 Overview of the Approach

In this subsection, we present an overview of our approach, including the flowchart of the approach and the introduction of the main steps, and then explain each step

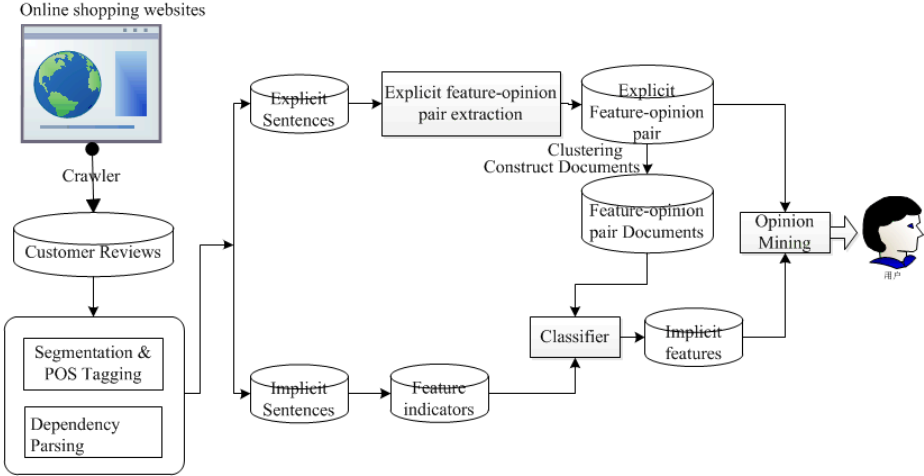


Fig. 1. Framework of Our Approach

in details. As it has been shown in Fig. 1, it takes the corpus of customer reviews that have been crawled from the online shopping websites as input, and generates the opinion mining summary consisting of both explicit feature-opinion pairs and implicit feature-opinion pairs as the output.

Our approach is consisted of three main steps, including the explicit feature-opinion pair extraction, feature-opinion pair training document construction and implicit feature identification. The detail of each step is described in the following subsections.

3.3 Explicit Feature-Opinion Pair Extraction

There are many existing research works on feature-level opinion mining that are dedicated to extract explicit feature-opinion pairs from customer reviews. Some supervised learning models, such as HMM(Hidden Markov Model) model and CRF(Conditional Random Fields) model, and topic models, such as the MaxEnt-LDA (a Maximum Entropy and LDA combination)[23] hybrid model, are widely used in this task. In this paper, we propose a rule based method to extract feature-opinion pairs from review sentences.

Rule Based Method. This method exploits Chinese dependency grammar to extract feature-opinion pairs. Firstly, we use Chinese dependency grammar to set several rules. Then we make use of these rules to extract candidate feature-opinion pairs. In order to improve the precision of feature-opinion pair extraction, we construct the candidate feature word set \mathcal{CF} and the candidate opinion word set \mathcal{CO} for each product. Since adjectives are quite likely to be opinion words and nouns are likely to feature words, so we extract adjectives and nouns from review sentences, and consider adjectives as candidate opinion words and nouns

as candidate feature words. In addition, we use a stopword list to filter many product-irrelevant adjectives and nouns from the candidate set \mathcal{CO} and \mathcal{CF} . Some frequently used non-adjective opinion words and non-noun feature words are also supplemented.

After analyzing and studying the dependency parsing results of review sentences, we find that most of the discussed features are in subject-predicate(SBV) structure or DE (“的”) structure. Therefore, we mainly exploit the two kind of dependency relation as rules to extract feature-opinion pairs. According to our observation, when the feature word appears before the opinion word, the feature usually satisfy the SBV relation with the opinion word, and when the feature appears after the opinion word, there usually exist a DE structure between the feature and the opinion word. Based on the above observations, Three different rules have been defined to tackle different types of sentence structures to extract the explicit feature-opinion pairs. A summarized representation of these rules is presented in the following paragraphs.

Rule-1: In a dependency relation SBV, the dependency structure is denoted as $sbv(w_1, w_2)$, which means that word w_2 depends on word w_1 through SBV, if word w_2 belongs to the opinion word set \mathcal{CO} and word w_1 belongs to the feature word set \mathcal{CF} , then $\langle w_1, w_2 \rangle$ can be extracted as feature-opinion pair.

Rule-2: In a dependency relation SBV, the dependency structure is denoted as $sbv(w_1, w_2)$, which means that word w_2 depends on word w_1 through SBV, if word w_1 belongs to the feature word set \mathcal{CF} and word w_2 doesn't belong to the opinion word set \mathcal{CO} , and after word w_2 there is a word w_3 that belongs to the opinion word set \mathcal{CO} , then $\langle w_1, w_3 \rangle$ can be extracted as feature-opinion pair.

Rule-3: In a dependency relation DE, if there is a word w_1 that belongs to the opinion word set \mathcal{CO} before the word “的” and a word w_2 belongs to the feature word set \mathcal{CF} after the word “的”, then $\langle w_2, w_1 \rangle$ can be extracted as feature-opinion pair.

The process of the rule based method is described in algorithm 1. We exploit the word segmentation, part-of-speech(POS) tagging and dependency parsing to process the customer reviews. Based on the results of the preprocessing, it can be easy to judge whether a sentence satisfied a rule.

3.4 Feature-Opinion Pair Training Document Construction

If we regard each explicit sentence as a training text, then the topic or category of this sentence can be labeled as the sentence's feature-opinion pair. For example, for the explicit sentence “送货很快，早上下单下午就送到了!” (Delivering is very fast, order in the morning and have received in the afternoon!), the feature-opinion pair $\langle \text{“送货”, “快”} \rangle$ ($\langle \text{delivering, fast} \rangle$) can be viewed as the sentence's labeled topic. If a sentence s_k contains more than one feature-opinion pair (FO_k denotes the sentence s_k 's feature-opinion pair set), then the sentence can be classified into each feature-opinion pair topic of FO_k .

Algorithm 1. Rule based algorithm

Input: Review sentences in the corpus.**Output:** A set of feature-opinion pair.

- 1: **for** each sentence s_k in corpus **do**
 - 2: **for** each rule RL_i in the rule set \mathbb{R} **do**
 - 3: **if** match the rule RL_i **then**
 - 4: extract the feature-opinion pair $fo = \langle f, o \rangle$, put fo into the sentence s_k 's feature-opinion pair set FO_k
 - 5: **end if**
 - 6: **end for**
 - 7: **end for**
 - 8: return the set of feature-opinion pair.
-

Feature-Opinion Pair Clustering. For each opinion word, there are usually more one feature-opinion pair that contains the opinion word. For a feature-opinion pair $\langle f, o \rangle$, it means that the opinion word o is used to describe the feature word f . In review sentences, a opinion word generally can be used to describe several different features. For example, the opinion word “好” (good) is often used to describe a lot of product features, such as “手机” (mobile phone), “屏幕” (screen), or “质量” (quality). In product reviews, many different feature words or phrases may be used to express the same feature. For example, features “音质” (vocality quality), “音乐” (music) and “音效” (sound effect) are all related to the same product feature “声音” (vocality). So for each opinion word o , we cluster the feature-opinion pairs $\mathcal{FO}(o) = \{\langle f_1, o \rangle, \langle f_2, o \rangle, \dots, \langle f_n, o \rangle\}$ that contains the opinion word based on the conceptual and semantical relation of these features $\mathcal{F}(o) = \{f_1, f_2, \dots, f_n\}$. Our clustering method is based on [22], we mainly exploit the sharing words and the lexical similarity to cluster features. The size of the feature set $\mathcal{F}(o)$ is relatively small compared with the whole set of features \mathcal{F} , so it is much easier and more effective to the clustering of feature-opinion pairs.

After getting the set of clustered feature-opinion pair, we construct the training document for each clustered feature-opinion pair. For each clustered feature-opinion pair, we collect the sentences that contain the feature-opinion pair into a document, which is labeled by the clustered feature-opinion pair. The document constructing process is presented in algorithm 2.

3.5 Implicit Feature Identification

By constructing the feature-opinion pair training set, the problem of identifying implicit features can be formulated into a text classification problem. Thus many existing text classification approaches can be used to solve this problem. In this step, we mainly deal with the implicit sentences. For each implicit sentence $\mathcal{I}S_k$, the set of opinion word can be denoted as $\mathcal{I}O_k = \{o_1, o_2, \dots, o_n\}$. The task of implicit feature identification is to find the implicit feature f_i for each opinion word o_i in $\mathcal{I}O$. The set of clustered feature-opinion pair that contains opinion word o_i can be denoted as $\mathcal{FO}_c(o_i) = \{\langle f_{c1}, o_i \rangle, \langle f_{c2}, o_i \rangle, \dots, \langle f_{cm}, o_i \rangle\}$. The key

Algorithm 2. Document construction

Input: The set of feature-opinion pair of explicit sentences.**Output:** A set of feature-opinion pair document.

- 1: **for** each opinion word o_i in the set of opinion word O **do**
 - 2: get the $\mathcal{FO}(o) = \{ \langle f_1, o \rangle, \langle f_2, o \rangle, \dots, \langle f_n, o \rangle \}$
 - 3: cluster the feature-opinion pairs
 - 4: get the clustered feature-opinion pairs $\mathcal{FO}_c(o_i) = \{ \langle f_{c1}, o_i \rangle, \langle f_{c2}, o_i \rangle, \dots, \langle f_{cm}, o_i \rangle \}$
 - 5: **end for**
 - 6: **for** each explicit sentence s_k in corpus **do**
 - 7: **for** each feature-opinion pair $\langle f_i, o_i \rangle$ in FO_k **do**
 - 8: put the sentence into the document $d(f_{c_i}o_i)$ of the clustered feature-opinion pair $f_{c_i}o_i = \langle f_{c_i}, o_i \rangle$ that includes $\langle f_i, o_i \rangle$
 - 9: **end for**
 - 10: **end for**
 - 11: return the set of clustered feature-opinion pair document \mathcal{D} .
-

issue of this problem is to find the feature f_{c_i} that the opinion word o_i in implicit sentence $\mathcal{I}S_k$ has modified from the feature set $\mathcal{F}_c(o_i) = \{f_{c1}, f_{c2}, \dots, f_{cm}\}$. As the feature-opinion pair can be regarded as the sentence's topic or category, thus the problem of finding the implicit feature f_{c_i} for opinion word o_i in implicit sentence $\mathcal{I}S_k$ has been transformed into a text classification problem.

In order to classify the implicit sentence with a opinion word o_i into the most probable feature-opinion pair $\langle f_i, o_i \rangle$ topic, we design a topic-feature-centroid classifier based on [2]. We modify the centroid construction and classification process to accommodate the situation in this problem.

Topic-Feature-Centroid Construction: Different from the centroid-based approaches in [2] that uses all the words in the corpse to form the lexicon set, we use only a small set of feature-related discriminative words in the training set to construct the lexicon set. For instance, considering the collected training set of feature-opinion pair $\langle \text{“送货”}, \text{“快”} \rangle$ ($\langle \text{delivery}, \text{fast} \rangle$), there are over two hundred different words in this topic, while only a very small number of words contribute to the feature space of this topic, such as word “速度” (speed), “下单” (order) and so on. Many other words, such as “很” (very), “是” (is), “有” (have) and so on, even some of them with a very high frequency, hardly have any discrimination for this topic. Moreover, such irrelevant words could bring on a lot of noise in the representation of the topic. Therefore, in the construction of the lexicon set, we only consider nouns, adjectives and verbs in the training set, and we also construct a filter word set to remove the stop words and irrelevant words. The constructed lexicon set is denoted as $\mathcal{L} = \{wf_1, wf_2, \dots, wf_L\}$, so the centroid for category $\langle f_j, o_j \rangle$ can be represented by a word vector $Centroid_j = \{wf_{1j}, wf_{2j}, \dots, wf_{Lj}\}$, where $wf_{kj} (1 \leq k \leq L)$ represents the weight for word wf_k .

In our topic-feature-centroid classifier, we derive a different formulation for the calculation of the weight for word wf_k . The weight for word wf_k of topic $\langle f_j, o_j \rangle$ is calculated as following:

$$wf_{kj} = f_{w_k} \times \log\left(\frac{|C|}{|CF_{w_k}|}\right) \quad (1)$$

where f_{w_k} is the word w_k 's frequency in the training document of feature-opinion pair topic $\langle f_j, o_j \rangle$, $|c|$ is the total number of feature-opinion pair topics for the given opinion word o_j , $|CF_{w_k}|$ is the number of feature-opinion pair topics that contains the word w_k . When a word w_k occurs in every feature-opinion topic, the value of wf_{kj} is 0 because $\log\left(\frac{|C|}{|CF_{w_k}|}\right)$ becomes 0, which means that word w_k has no discrimination for the topic. Thus our weight calculation method can produce more discriminative features for the feature-opinion topic.

Classification: After the centroid vector of each category is obtained, the implicit sentence is classified by using a denormalized cosine measure:

$$C' = \arg \max_j (\vec{s}_i \bullet \overrightarrow{Centroid_j}) \quad (2)$$

where \vec{s}_i is the word vector representation for the implicit sentence $\mathcal{I}S_k$, since the sentence is usually very short, so we only concern the word's appearance or not. We use the binary representation to denote the word's weight in \vec{s}_i . By using this denormalized cosine measure, it preserves the discriminative capability of feature-opinion pair topic's centroid vector. Since the size of the vector space here is relatively small, so the denormalized measure can be more discriminative for the classification.

The process of implicit feature identification is described in algorithm 3.

Algorithm 3. Implicit feature identification

Input: The set of implicit sentences.

Output: A set of implicit feature-opinion pair.

- 1: **for** each implicit sentence $\mathcal{I}S_k$ **do**
 - 2: **for** each opinion word o_i in $\mathcal{I}O$ **do**
 - 3: apply the topic-feature-centroid classifier for o_i
 - 4: get the implicit feature fc_i
 - 5: **end for**
 - 6: **end for**
 - 7: return the set of implicit feature-opinion pair.
-

4 Experiments

In this section, we conducted several experiments and evaluate the performance of our approach. Firstly, we describe the data sets used in our experiments. Then we give the definition of several performance metrics. Lastly, experiment results and corresponding analysis are described. Both the results of explicit feature-opinion pair extraction and implicit feature identification have been evaluated.

4.1 Date Sets and Evaluation Measures

Since there is no standard data set for our experiment, so we crawled the experiment data from the popular Chinese shopping website, Amazon.cn¹, the regional website of Amazon.com in China. Customer reviews are collected from two different domains: cell phone and digital camera. There are totally 4083 reviews and 12760 sentences in our data set. Both the explicit feature-opinion pair and implicit feature-opinion pair of each sentence are manually annotated by two research students in our lab. To be fair, those sentences that are annotated inconsistently have been removed and the rest has been confirmed by the author. The details of the data sets are given in Table 1.

Table 1. Experiment Data

Data Sets	Reviews	Sentences	Explicit features	Implicit features
Cell Phone	2694	8305	4233	1449
Digital Camera	1389	4455	1817	798
Total	4083	12760	6050	2247

The traditional precision (P), and recall (R) and F-measure (F) have been used to evaluate our experiment results of both explicit feature-opinion pair extraction and implicit feature identification. The F-measure is defined as follows:

$$F = \frac{2 * precision * recall}{precision + recall} \quad (3)$$

4.2 Evaluation of Explicit Feature-Opinion Pair Extraction

An important step of our approach is to extract the explicit feature-opinion pairs from explicit sentences. The construction of the training document is based on the result of the extracted explicit feature-opinion pairs. In this paper, we use LTP² to accomplish the Chinese word segmentation, part-of-speech(POS) tagging and dependency parsing.

Table 2 shows the result of explicit feature-opinion pair extraction by using our rule based method. As we can see from the table, our rule based method achieves a comparatively satisfactory result in the extraction of feature-opinion pairs. In the construction of the candidate feature word set \mathcal{CF} and the candidate opinion word set \mathcal{CO} for each product, we consider nouns as candidate features and adjectives as candidate opinion words, and also add some verbs to complement the candidate word set. For example, the verb “送货” (deliver) is frequently used as feature and the verb “喜欢” (like) is frequently used as opinion word in customer reviews. A filter word list was constructed to remove many product-irrelevant nouns and adjectives from the candidate set, such as “朋友” (friends), “伤心” (sad) and so on.

¹ <http://www.amazon.cn>

² <http://ir.hit.edu.cn/ltp/>

Table 2. Result of explicit feature-opinion pair extraction

Data Sets	Precision	Recall	F-measure
Cell Phone	80.21%	79.99%	80.10%
Digital Camera	81.95%	83.43%	82.68%

4.3 Evaluation of Implicit Future Identification

In the end, we give the final experimental results via using our proposed classification-based approach. Our classification-based approach is compare with the rule based approach coAR[3]. We implement the approach coAR proposed in [3]. The best results for each approach is listed in Table 3.

Table 3. Result of implicit feature identification

Data Sets	Our Approach			CoAR		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Cell Phone	82.07%	68.48%	74.66%	67.88%	52.93%	59.48%
Digital Camera	85.59%	72.93%	78.76%	79.94%	66.91%	72.85%

CoAR mined the association rules of explicit feature-opinion pair based on the co-occurrence of opinion word and feature word, and then used the clustered rules to identify the implicit feature for a given opinion word.

As it can be observed from Table 3, our approach outperforms coAR on all the evaluation metrics for corpora in both cell phone data set and digital camera data set. Co-AR used the mined association rules from the review corpus to identify the implicit feature for the given opinion word by rule matching, which means that their approach always map the opinion word to the same feature word without considering the context of the implicit sentence. While our classification-based approach not only exploit the association relations by extracting explicit feature-opinion pairs, but also take into account the context of the implicit sentence by using the category-feature-centroid classifier to map the opinion word in a specific implicit sentence to the most probable feature word. Thus our approach’s precision is higher than coAR. In addition, our approach’s recall is also higher than the coAR approach. This is because that the rule based coAR approach only adopted the association rules whose weight is greater than the threshold as robust rules. The higher threshold can weed out the lower-frequency association rules and promote the precision, but it would reduce the recall. No matter what threshold is selected, it can not capture a significant number of uncommon association rules. This shortcoming of the rule based approach determines that the recall of coAR is limited to a certain extent.

5 Conclusion and Feature Work

In this paper, we propose a novel classification-based approach to deal with the problem of implicit feature identification. By constructing the document for the

clustered feature-opinion pair, the training document that has been labeled by the specific clustered feature-opinion pair can be obtained. Then the problem of implicit feature identification has been formulated into a text classification problem. A rule based method has been proposed to extract explicit feature-opinion pairs from customer reviews. In the phase of implicit feature identification, a topic-feature-centroid classifier has been designed to perform the classification task. It should be pointed out that other feature-opinion pair extraction methods and text classification methods can also be used in our approach. Compared with the rule based approach coAR, our approach overcomes the shortcomings and limitations of the rule based approach and achieves a much better performance on all the measure metrics. However, some undesirable errors still exist in the result of implicit feature identification. Some are caused by the incorrect classification, some are caused by the wrong identification of implicit feature indicators. In our future work, we will explore the performance of approach by using several other text classification approaches in the implicit feature identification step.

References

1. Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G.A., Reynar, J.: Building a sentiment summarizer for local service reviews. In: WWW Workshop on NLP in the Information Explosion Era (2008)
2. Guan, H., Zhou, J., Guo, M.: A class-feature-centroid classifier for text categorization. In: Proceedings of the 18th International Conference on World Wide Web, pp. 201–210. ACM (2009)
3. Hai, Z., Chang, K., Kim, J.-J.: Implicit feature identification via co-occurrence association rule mining. In: Gelbukh, A.F. (ed.) CICALING 2011, Part I. LNCS, vol. 6608, pp. 393–404. Springer, Heidelberg (2011)
4. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
5. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: Proceedings of the National Conference on Artificial Intelligence, pp. 755–760. AAAI Press, MIT Press, Menlo Park, Cambridge (2004)
6. Hu, M., Liu, B.: Opinion feature extraction using class sequential rules. In: AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs, Palo Alto, USA (2006)
7. 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, pp. 1035–1045. Association for Computational Linguistics (2010)
8. Kobayashi, N., Inui, K., Matsumoto, Y.: Extracting aspect-evaluation and aspect-of relations in opinion mining. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1065–1074 (2007)
9. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, pp. 342–351. ACM (2005)

10. 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, pp. 115–124. Association for Computational Linguistics (2005)
11. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2), 1–135 (2008)
12. 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)
13. Popescu, A.-M., Etzioni, O.: Extracting product features and opinions from reviews. In: *Natural Language Processing and Text Mining*, pp. 9–28. Springer (2007)
14. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 105–112. Association for Computational Linguistics (2003)
15. Su, Q., Xiang, K., Wang, H., Sun, B., Yu, S.: Using pointwise mutual information to identify implicit features in customer reviews. In: Matsumoto, Y., Sproat, R.W., Wong, K.-F., Zhang, M. (eds.) *ICCPOL 2006. LNCS (LNAI)*, vol. 4285, pp. 22–30. Springer, Heidelberg (2006)
16. Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., Swen, B., Su, Z.: Hidden sentiment association in Chinese web opinion mining. In: Proceedings of the 17th International Conference on World Wide Web, pp. 959–968. ACM (2008)
17. 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)
18. Wang, W., Xu, H., Wan, W.: Implicit feature identification via hybrid association rule mining. *Expert Systems with Applications* (2012)
19. Wiebe, J.M., Bruce, R.F., O’Hara, T.P.: Development and use of a gold-standard data set for subjectivity classifications. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, pp. 246–253. Association for Computational Linguistics (1999)
20. Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., Patwardhan, S.: Opinionfinder: A system for subjectivity analysis. In: Proceedings of HLT/EMNLP on Interactive Demonstrations, pp. 34–35. Association for Computational Linguistics (2005)
21. Wilson, T., Wiebe, J., Hwa, R.: Just how mad are you? Finding strong and weak opinion clauses. In: Proceedings of the National Conference on Artificial Intelligence, pp. 761–769. AAAI Press, MIT Press, Menlo Park, Cambridge (2004)
22. Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 347–354. ACM (2011)
23. Zhao, W.X., Jiang, J., Yan, H., Li, X.: Jointly modeling aspects and opinions with a maxent-lda hybrid. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 56–65. Association for Computational Linguistics (2010)
24. Zhuang, L., Jing, F., Zhu, X.-Y.: Movie review mining and summarization. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 43–50. ACM (2006)

Role of Emoticons in Sentence-Level Sentiment Classification

Martin Min, Tanya Lee, and Ray Hsu

Netbase Solutions, Inc., Mountain View, USA
{cmin, tlee, rhsu}@netbase.com

Abstract . Automated sentiment extraction from social media is enabling technology to support gathering online customer insights. The basic sentiment extraction is semantic classification of a text unit as positive or negative using lexical and/or contextual clues in a natural language system. From the input side, it is observed that social media as a sub-language often uses emoticons mixed with text to show emotions. Most emoticons, e.g. :=), are not natural language words, but textual symbols using characters to present a smiley face. Intuitively, such symbols are innately associated with emotions, whether *happy*, *annoyed* or *don't care*, hence important clues for helping sentiment classification. Previous research has involved the limited use of emoticons as noisy labels in sentiment learning but detailed study on how noisy or useful they are has not been done. This paper presents a comprehensive data analysis study of the role of emoticons in sentence level sentiment classification. Various investigations are conducted on a fairly large annotated social media corpus, selected by our consumer insight analytics system. This corpus consists of 40,548 sentiment-rich sentences which business users are truly interested in mining. The study shows that the consistency between positive/negative emoticons with human judgment in this corpus is as high as 75.2%. Another larger randomly selected corpus consisting of 300,000 sentences from social media shows its consistency with human judgment to be 40.1%. A further study finds that emoticons' recall contribution to sentiment classification is moderate, nevertheless, the data containing emoticons and brands are guaranteed to be quality social media representing customers' voice instead of businesses' voice such as press news. In addition, emoticon is an additional factor to help extract sentiments where other linguistic clues are insufficient.

Keywords : emoticon, sentiment extraction, sentiment classification, customer insight, social media.

1 Introduction

The rapid growth of social data from such online social networks as Twitter and Facebook has aroused enormous interest in the mining and discovery of customer insights, recognized as significant for businesses. For instance, Twitter has over 500 million registered users as of 2012, generating over 340 million tweets daily, which is equivalent to 3,935 tweets per second. Sina Weibo, a Twitter like micro-blog service in

China, has an active user base of over 40 million. Such huge amounts of unstructured text data involve enormous amount of customer voice that is invaluable to leading brands and businesses, especially in marketing and sales departments. They need to monitor, react, engage, and publish at the speed of social in real time in order to compete. The main motivation behind the adoption of social media intelligence tools like ours lies in the fact that people are increasingly sharing their opinions on products and services they buy or want to buy on social networks. Recent estimates indicate that on average one in every three blog posts and one in five tweets involve comments on products, services or brands (Hogenboom et al. 2013). Netters freely talk about whether they love or hate a brand, and lots of times they compare it with other brands in the same category. Apparently, such information would be really important for businesses to keep track of consumers' attitudes toward their brands and the management can make faster decisions based on the social intelligence when it is extracted from the huge social data pool and analyzed properly.

Sentiment analysis is an essential part of a commercial social media intelligence platform. The majority of sentiment analysis systems are machine learning based, taking a traditional text classification approach to train models like Naïve Bayes, Maximum Entropy or Support Vector Machine. The text unit for classification is usually a document or paragraph consisting of multiple sentences. Typical examples of such data are movie or product reviews. When trained on domain data, those classifiers generally achieve over 80% accuracy for coarse-grained sentiment classification, as positive, negative or neutral, as reported in the study (Pang and Lee 2008). While machine learning based sentiment classification works well in domain data at document or paragraph level, it faces challenges in handling object focused short messages (e.g. twits) in a commercial sentiment analysis system today when the dominant social media platform is mobile and the posts tend to be shorter from mobile users.

Social media as a sub-language is sometimes full of emoticons mixed with text to show emotions of the poster. Most emoticons, e.g. :=), are not natural language words, but textual symbols using characters. Among various definitions, Wikipedia defines emoticon as “a meta-communicative pictorial representation of a facial expression ... draw a receiver's attention to the tenor or temper of a sender's nominal verbal communication, changing and improving its interpretation”. It mainly uses a combination of punctuation marks to mimic a smiley face to express a person's feeling, mood or intention. Intuitively, people would assume that such symbols are innately associated with emotions, whether *love*, *hate* or *don't care*, hence important clues for helping sentiment classification. With the rapid growth of social media uses globally, it's generally recognized that emoticons have been playing an increasingly important role in online communications, especially for the younger generation. In light of this, a natural language system built for sentiment analysis is expected to take advantage of this special linguistic phenomenon, which is typically not in the scope of the grammar or vocabulary research of a language.

Unlike the main stream sentiment analysis based on statistical models using machine learning, we have built a rule-based high precision sentiment analysis system based on a parser for use in the product of mining consumer insights from social media. This system is designed to address two major challenges of machine learning systems: (i) brand focused sentence-level sentiment classification for short messages; (ii) extracting reasons behind sentiments to answer why questions. The research on emoticons reported in this paper is motivated by the need to enhance the system in handling (i) in the context of mining customer sentiments towards a brand. But the analysis and experiments we have done also serve the purpose of enlightening the researchers in

both machine learning world and the grammar world with better understanding of the role of emoticons in sentiment analysis. In fact, it reveals a pitfall facing some earlier researchers (Read 2005; Zhao et al. 2012) who assume emoticons are handy and reliable sentiment indicators and therefore use them to collect training corpus for sentiment classification.

The major contribution of this study lies in the fairly comprehensive study of emoticon's distribution in social media and its role in a sentiment analysis system. We aim to accomplish two specific goals.

- Provide a statistical analysis of how emoticons are used in social media from various perspectives
- Provide an evaluation of emoticon's roles in our brand-focused sentence level sentiment classification system by calculating its precision and recall impact on system performance.

The remainder of the paper is organized as follows. Section 2 reviews related work in using emoticon as a clue for sentiment classification. Section 3 briefs our parsing-supported sentence-level deep sentiment system to provide a background for this study. In Section 4, emoticon-related experiments are set up and results are presented, focusing on emoticon's statistical distribution in social media, and its precision and recall impact on data quality. We present our findings and conclusions in Section 6.

2 Related Work

The popularity of emoticons (or smileys) comes hand in hand with the growth of social network. They have been extremely popular in social media among the younger generation and the seasoned netters. Despite their use everywhere in the online text, linguistically, they are not a "legitimate" part of natural language vocabulary or morphology, hence belonging to so-called Unnatural Language Processing (UNLP, Ptaszynski et al. 2011). Some emoticons are fairly universal as symbols of emotion, and others are language dependent. Survey shows that they are the second most important vehicles for expressing emotions in online communication (Ptaszynski et al. 2011).

In the context of NLP, the use of emoticons has attracted machine learning researchers for the sentiment classification. Emotions seem to be a handy and reliable indicator of emotions and hence are used either to help automatically generate a training corpus for sentiment classification or to act as *seeds* or one type of evidence features to enhance sentiment classification (Davidov, Tsur and Rappoport 2010; Liu, Li and Guo 2012; Read 2005; Zhao et al. 2012; Yang, Lin and Chen 2007; Hogenboom1 et al. 2013).

Not much has been done in evaluating the contributions of emoticons in sizable real life social media corpora, in the context of brand-centered sentiment analysis. That is one major motivation and value for this study.

Ptaszynski et al. (2011) proposes that emoticon research consists of four lines of tasks: (1) detection; (2) extraction; (3) parsing; (4) semantic analysis; (5) generation; (6) evaluation. Our work involves (1), (2) and (6). The work involved in (3), (4) and (5) assumes the productive nature of emoticons, similar to the open morphology study in natural languages. For the following reason, at least for English, this is not a real issue.

There are thousands of varieties of emoticons due to the semantic compositionality of its components, in ways that are very close to flexible word formation in some natural language morphology, with a smiley face being made with various types of eyes, nose and mouth etc. (Strapparava and Mihalcea 2008). However, our study demonstrates that the frequency distribution of different emoticons is very different, and many theoretically possible combinations do not really add to the system due to the infrequency of their appearance in data. At least for English, the top n ($n < 1000$) emoticons are easily listable in lexicon and can fulfill the identification of emoticons with very high precision and recall.

3 Sentence-Level Sentiment Classification

Our sentence-level sentiment system is supported by a natural language parser we developed. Each incoming sentence is analyzed by the full NLP parser, starting from tokenization, POS (Part-of-Speech) tagging, chunking and ending in decoding a dependency parse tree enriched with various syntactic and semantic features, on top of which deep sentiment extraction capability is built. For instance,

I like the camera of iPhone because the photo quality is higher.

From this sentence the system is able to extract information as follows:

Sentiment: *positive*
 Object: *iPhone*
 Aspect: *camera*
 Reason: *photo quality (higher)*

The concept of deep, fine-grained sentiments is proposed in contrast to the dominant practice of shallow, course-grained sentiment analysis, thumbs-up and down (or plus neutral) classification, coupled with sentiment association based on proximity. This concept is inspired by the needs from the real world market analysts who were initially very happy with the precision of our sentiment insights and later told us they need more actionable insights and hope we can answer the why questions with regards to sentiments. Over time, the deep sentiments evolve in the process of engaging the users of customer insights and become fairly mature as a standard to drive the research and development supported by deep parsing. To shed some light in the process, a deep sentiment system should be able to extraction insights that can answer these questions in addition to the sentiment classification insight:

- Which brand is this sentiment about? (association insight)
- Can the system associate sentiments not only with a brand such as iPhone, but also with a feature of the brand, say, screen? (granularity insight)
- Who made the sentiment comment? (customer background insight)
- How intense is the sentiment? (passion intensity insight)
- Finally, most important of all, what is the reason of the sentiment? (why insight)

Systems that can answer such questions provide invaluable actionable insights to businesses. For instance, it is much more insightful to know that consumers love the

online speed of iPhone 4s but are very annoyed by the lack of support to flash. This is an actionable insight, one that a company could use to redirect resources to address issues or drive a product's development. Extraction of such insights is enabled by our deep parsing.

Since our sentiment extraction must be object centered, meaning that the sentiment extracted must be toward a topic or an object mentioned in a sentence, which could be a brand, or person or location name. We typically select 10-20 brands to evaluate the system. For instance, in the most recent release the brands we used are:

iPhone, Walmart, Listerine, Costco, Olive Garden, Taco Bell, Tylenol, Camaro, Prius, Ikea, JetBlue, Skype, Yoplait, Playstation, fish oil, Pepsi.

We use CrowdFlower's anonymous annotation service to annotate the data, and 75% or above inter-annotator agreement with at least four judges each time is used for benchmarking. The precision we have achieved is 87% on average across the brands.

Due to the fact that the sentiment is generally sparse in randomly selected data, we have not really taken a standard approach to evaluate the recall, since that would require annotating a significant amount of data than precision evaluation. Also, our experiences have shown that to a certain point, increasing recall is an incremental process, especially when taking into consideration multiple domain factors. Thus, our evaluation has been focused on precision benchmark. For tracking relative recall, we simply measure the total number of extracted sentiment mentions and their percentage given a certain amount data processed by the system.

4 Experiments and Results

In this section, we present three experiments aiming to answer three questions:

- How emoticons are generally used and statistically distributed in social media?
- What's the precision of sentiment classification when only using emoticon as evidence?
- What's recall contribution of emoticons to sentiment classification?

These are questions that can help drive the design and development in properly involving emoticons in a sentiment system. It needs to be noted that in our approach to brand-focused, sentence-level classification, the identification of sentiment from a sentence must be targeted at a specific object.

A. Experiment Setups

One of the major resources in the experiments is the emoticon lexicon with a total of 1258 entries, which we collected during the system development from social data. We manually marked 106 entries as positive indicator and 233 as negative indicator, leaving the rest as unspecified.

In addition to the emoticon lexicon, we use two corpora for the evaluation in the experiment. The first one is a human annotated corpus with 40,548 sentences, which is an accumulation of some of the data which our QA department prepared for system evaluation. Each sentence is annotated by four annotators, and a 75% agreement among annotators is the cutoff threshold we adopt for an agreement. To ensure the objective evaluation, our QA department uses a crowd sourcing service for the

annotation so the human judges have no association whatsoever with the development team. Each sentence is annotated with a sentiment choice of positive, negative or neutral towards the corresponding object associated with the sentiment.

Another aspect of this corpus is that, sentences in this corpus are not randomly collected, but selected from our sentiment analysis system's output. The implication of this choice is that the data must be sentiment-rich, meaning that many sentences from this corpus would be either positive or negative due to the fact that it is a much smaller subset of data that has been filtered by our sentiment extraction system. One of the main practical reasons for this is that, the primary use of this corpus is for evaluating system's precision, not recall, and the standard precision metric is measured by the fraction of extracted sentiments that are correct against human annotations. For precision measurement, this selection of data is good enough and as a matter of fact, much better. This is because, if the data is randomly selected, the majority of the random data would not contain any sentiment and it requires a much bigger size of data to be annotated. That is not only costly but also meaningless for precision evaluation. Our empirical study has shown that about 15%-25% of data contains a sentiment, depending on a specific brand. Given the fact that this corpus is selected in a way that it is guaranteed to be sentiment rich, we name it SelectCorpus.

As opposed to the SelectCorpus, we also have a second corpus, RandomCorpus, made up of randomly collected data from our content store. The only requirement for this corpus is that each sentence selected must contain at least one brand term, e.g. Listerine. Unlike the sentiment-rich SelectCorpus, the sentiment is much sparser in RandomCorpus. As discussed earlier, on average only 15%-25% of sentences are expressing a sentiment, either explicit or implicit. From a system evaluation point of view, such a random corpus is more representative of social media content. Hence, we would like this corpus to be another set of evaluation data used in the experiments.

While SelectCorpus is human annotated with sentiment, RandomCorpus is not, and thus its size is much larger, with a total number of 300,000 sentences, as opposed to 40,548 sentences from SelectCorpus. Without annotation, how would we measure precision and recall based on this corpus? Here we are not seeking for measuring the absolute performance in the traditional sense. Instead, we estimate precision and recall through these two formulas:

$$\text{Estimated Precision} = \frac{\text{count of agreed sentiment with emoticon} * 0.87}{\text{total count of extracted sentiment}}$$

$$\text{Estimated Recall} = \frac{\text{count of agreed sentiment with emoticon} * 0.87}{\text{total count of sentences} * 0.2}$$

In the formulas, 0.87 is the system's average precision score across brands, which is obtained through our series of evaluation over the system development course. Likewise, 0.2 is the average sentiment richness score in the range of 0.15-0.25 obtained for different brands in our evaluation. Sentiment Richness is defined as the ratio of the number of sentences containing sentiment versus the total number of sentences.

B. Experiment Results

Experiments are conducted and results are reported from two perspectives in evaluating emoticons: (i) statistical distribution of emoticon uses in social media; (ii) emoticon's impact on precision and recall for sentiment extraction.

i) Statistical distribution of emoticon uses in social media

We primarily use RandomCorpus to analyze emoticon's frequencies and distribution, since it is much larger than SelectCorpus. Table 1 lists overall frequencies and distribution in the corpus.

Table 1. Emoticon Sentiments in RandomCorpus

	counts	Percentile	Sent. count	Richness
Positive	6,752	65.4%		
Negative	1,314	12.9%		
Neutral	2,244	21.7%		
Total	10,310	100%	300,000	3.4%

Like the Sentiment Richness, the Emoticon Richness is defined as the ratio of the number of sentences containing an emoticon versus the total number of sentences in the corpus. This metric informs us of the overall frequency of emoticons used in social media and their maximum possible impact on a sentiment extraction system.

Given the over 1000 emoticons in our emoticon lexicon, we are curious about how each of these emoticons is actually used in social media. For this purpose, we count the frequency of each emoticon's use and its expressed emotion or mood based on the RandomCorpus. The top 10 used emoticons are listed in Table 2.

Table 2. Top 10 emoticons in RandomCorpus

Rank	Emoticon	Frequency	Percent	Emotion
1	:)	3,505	34.00%	Happy
2	:D	1,273	12.35%	Laugh
3	:(927	0.89%	Sad
4	;))	773	7.50%	Wink
5	:-)	711	6.86%	Happy
6	:P	433	4.21%	Tongue out
7	=)	319	3.10%	Happy
8	(:	309	3.00%	Happy
9	;-)	226	2.19%	Wink
10	XD	175	1.70%	Grin
Total	N/A	8,651	83.90%	N/A

Table 3 demonstrates that although the number of emoticons in social media is big, only a handful of emoticons are used far more frequently than all others.

Table 3. User counts of Unique Emoticon

Unique users	Number of users	Percentile
1	238000	95.02%
2	6325	2.53%
3	4350	1.74%
4	1265	0.55%
5	400	0.16%

ii) Emoticon’s impact on sentiment precision and recall

From Table 1, we see that the positive, negative and neutral ratio of emoticons used is 65.4%, 12.9% and 21.9%. This gives us the impression that most emoticons used are conveying positive emotions, either explicit or implicit. However, to what extent is using emoticons as a single sentiment clue correct, especially in our context of brand focused sentiment extraction? For instance, in the message “Someone asking a greeter at walmart to watch their child ----JOKE :) ha ha ha ha”, even though there is a positive emoticon indicating the poster is happy, there is no any indication of that sentiment is for the object “Walmart”. Our major goal here is to find out how reliable it is to use emoticon alone as a sentiment indicator.

We run two experiments on both SelectCorpus and RandomCorpus. The results are listed in Table 4 and Table 5, with the precision to be 75.2% from SelectCorpus and 40.1% from RandomCorpus.

Table 4. Precision Using Emoticon as Single Sentiment Clue Based on SelectCorpus

# of Pos agree	# of Neg agree	# of Pos as Neg	# of Neg as Pos	# of Neu as Pos	# of Neu as Neg	Precision
1052	88	211	84	56	25	75.2%

$$P = (1052+88) / (1052+88+211+84+56+25) = 75.2\%$$

Table 5. Precision Using Emoticon as Single Sentiment Clue Based on RandomCorpus

# of Pos agree	# of Neg Agree	# of Pos as Neg	# of Neg as Pos	# of Neu as Pos	# of Neu as Neg	Precision
4068	641	779	421	3001	1400	40.1%

$$P = [(4068+641) / (4068+641+779+421+3001+1400)] * 0.87 = 40.1\%$$

Each column in the tables represents:

- # of Pos(itive) Agree(ments): number of positive posts agreeing with positive emoticon mention
- # of Neg(ative) Agree(ments): number of negative posts agreeing with negative emoticon mention

- # of Pos(itive) with Neg(ative): number of positive posts with negative emoticon mention
- # of Neg(ative) with Pos(itive): number of negative posts with positive emoticon mention
- # of Neu(tral) with Pos(itive): number of neutral posts with positive emoticon mention
- # of Neu(tral) with Neg(ative): number of neutral posts with negative emoticon mention

The precision numbers differ significantly on the two corpora. It is not surprising however. Since the data in SelectCorpus is from our sentiment extraction system’s output, meaning that it is already filtered by our sentiment system. As a result, the sentiment of this data set is much richer. However, the precision number obtained from SelectCorpus still offers insights into how emoticon’s sentiment may overlap with the actual sentiment judged by human standards. The 75.2% precision seems to suggest that, when a sentence is guaranteed to be positive or negative, and when there is an emoticon in it, there is a fairly high chance of being the case that the emoticon can be trusted to be a fairly good sentiment indicator. In this case, the chance is 75.2%.

On the other hand, the 40.1% precision from RandomCorpus tells that in the real world, emoticon alone is not reliable enough to be taken as a sentiment dictator. As we can see from Table 5, there are a large number of neutral sentences that have been incorrectly classified as either positive (3001) or negative (1400) using emoticon. Mistaking neutral ones as positive or negative has been a common problem for a sentiment analysis system, and our study shows that emoticon cannot be immune to this headache either. For instance,

at the marriott hotel and i ran into vili, small world :D!

The highlighted *marriott* is the brand we were evaluating. Although the emoticon generally expresses some type of emotion, it does not really indicate a sentiment for the brand we are interested in.

Finally, we would like to get a sense of the recall contribution if emoticon is used as a sentiment indicator. The 3.4% Emoticon Richness score listed in Table 1 informs us that the maximum contribution to the system recall would be 3.4%, assuming *i*) none of the sentences containing an emoticon have been correctly classified by the system without using emoticon; *ii*) emoticon’s precision as an emoticon indicator is 100%. In reality, neither holds true though. As a result, the actual recall contribution would be lower. The result is presented in Table 6. The first column shows emoticons’ agreement with the annotation, column 2 is for the emoticons’ agreement with annotation missed by the system, column 3 is the recall improvement and column 4 shows the overall recall improvement.

Table 6. Emoticon’s impact on System Recall in SelectCorpus

	Agree w. key	System miss	Recall up	Average
Pos	1052	20	1.94%	2.72%
Neg	88	3	3.52%	

So how to assess the 2.72% recall improvement on SelectCorpus? The number does not seem to be impressive and significant. However, from a system development point of view, this improvement is meaningful. In our fairly mature English system, we have a total of 400+ rules, which consist of thousands of linguistic patterns built upon a semantic parser. However, the top ten mostly fired rules contribute to nearly 50% of all sentiments extracted. The vast majority of all other rules account for the long tail of the remaining 50% sentiments, where many individual rules contribute to less than 1%. These are either domain specific or linguistically specific rules. Table 7 lists the top ten fired rules in our system. Individual rules that contribute to little recall but can correct eye-catching errors cannot be ignored in a real life system. In this sense, the emoticon provides a low-hanging fruit for enhancing the data quality which should not be ignored either.

Table 7. Top ten Rules Contributing to Sentiment Extraction System

Firing Frequency	Positive	Negative
# 1	10.16%	9.89%
# 2	9.59%	7.12%
# 3	5.15%	6.11%
# 4	4.96%	5.22%
# 5	3.49%	3.93%
# 6	3.95%	3.42%
# 7	3.60%	3.11%
# 8	3.13%	2.73%
# 9	2.88%	2.50%
# 10	2.85%	2.25%
Total	49.76%	46.37%

However, it has to be noted that since data from SelectCorpus is not randomly sampled as discussed earlier, the actual number will be different from 2.72%, but the upper boundary would be 4.2% for RandomCorpus. More experiments will be needed in future to estimate the recall improvement using the formula “Estimated Recall” presented earlier, based on randomly selected data.

5 Conclusion and Future Work

We have performed a fairly comprehensive quantitative analysis of how emoticons are used in social media, how reliable it is to use emoticon alone as a sentiment trigger, and what could be the recall contribution of emoticon in a brand focused sentiment extraction system. We demonstrate that emoticon alone without considering other linguistic evidence is not sufficient to dictate a sentiment toward an object. Our study shows that its recall contribution in our context is not significant, but it is meaningful to help enhance linguistic and/or domain specific sentiment extraction. Ongoing and future work will be focused on what other linguistic factors could be used together with emoticon to improve precision and recall, such as sentence length, and other emotion-related lexical items

including many weak emotion words. In addition, the study of language-dependent part of emoticons, especially for the Eastern vs. Western distinction, is also interesting and would be beneficial to our multilingual program.

Acknowledgment. Thanks to our Quality Assurance team for preparing large testing corpora in which we are enabled to conduct the data analysis.

References

- [1] Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of COLING 2010: Poster Volume, Beijing, pp. 241–249 (2010)
- [2] Hogenboom, A., Bal, D., Frasincar, F., Bal, M., Jong, F., Kaymak, U.: Exploiting Emoticons in Sentiment Analysis. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 703–710 (2013)
- [3] Liu, K., Li, W., Guo, M.: Emoticon Smoothed Language Models for Twitter Sentiment Analysis. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
- [4] Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2(1), 1–135 (2008)
- [5] Ptaszynski, M., Rzepka, R., Araki, K., Momouchi, Y.: Research on Emoticons: Review of the Field and Proposal of Research Framework. *言語処理学会 第 17 回年次大会 発表論文集* (2011)
- [6] Read, J.: Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classification. In: Proceedings of the ACL Student Research Workshop, pp. 43–48 (2005)
- [7] Strapparava, C., Mihalcea, R.: Learning to identify emotions in text. In: SAC (2008)
- [8] Zhao, J., Dong, L., Wu, J., Xu, K.: MoodLens: an emoticon-based sentiment analysis system for Chinese tweets. In: KDD 2012, pp. 1528–1531 (2012)

Emotional McGurk Effect? A Cross-Cultural Investigation on Emotion Expression under Vocal and Facial Conflict

Aijun Li¹, Qiang Fang¹, Yuan Jia¹, and Jianwu Dang²

¹Institute of Linguistics, Chinese Academy of Social Sciences, Beijing, China
{liaj, fangqiang, jiayuan}@cass.org.cn

²Tianjin University, Tianjin, China
Dangjianwu@tju.edu.cn

Abstract. A multi-modal emotion perceptual experiment is conducted cross-culturally to investigate the difference of expressing and perceiving emotions across Chinese and Japanese. Focus is on the cultural effect on the interaction between the combination of vocal and facial expression and perception. In this paper, part of the perceptual results is reported for the AV-conflicting stimuli produced by a Chinese female speaker and perceived by Chinese and Japanese listeners. The results support the assumptions that (i) When listeners decoding the conflicting AV stimuli, they might rely on some modality more than another across different emotions. (ii) Although common psychological factor contributes to the emotional communication, the decoding of conflicting AV information will be affected by culture background, and (iii) the emotional McGurk effect exists, and it may also be related to cultural norms of the encoder/listener.

Keywords: Emotion, emotional McGurk effect, Cross culture, multimodality.

1 Introduction

Emotion theories and emotion psychology are currently concerned with the universality and cultural relativity of emotional expression, which, in fact, is a key issue exploring ‘what is the essence and function of emotion?’ It is generally acknowledged that emotional expression is both psychobiologically and culturally controlled, but the respective effect imposed by psychobiology and culture on emotional expression remains unexplored. The earliest predecessors studying cross-cultural emotion include Charles Darwin[1], Ekman[2] and Izard[3]. They notice that listeners from one culture have the ability to decode the facial expression of an actor from another culture. They claim that like decoding facial expression of emotion, people from different culture can decode vocal expression of emotion. Therefore, from the psychobiological perspective, emotion decoding is universal. Cross-cultural studies on emotion encoding and decoding are needed to supply speech technology with culturally-relative emotional expressions. Erickson[14] made a review on cross-linguistic studies, recently, lot of research have been carrying on cross-cultural research on emotional speech[4-11], some of the results are consistent such as the perception of emotional expressions is more successful

when the stimuli are multimodal, facial expression plays a major role in the correct decoding of emotion; some are inconsistent such as the recognition of the emotions was not influenced by cultural differences[11], while some thought there exist cross-cultural difference[4-6]. Besides, they suggest “Anger joy and sad” may constitute three basic emotions[10], and listeners from different cultures show different sensitive degrees to the acoustic parameters when decoding emotions.

Speech communication is a physiological process, conveying both audio and visual information. Human’s perception bases on the information transmitted by both channels. Generally, in speech communication, the information from two channels is complementary and coherent. But when the information is conflicting and nevertheless integrated then the percept in one of the modalities might be changed by the other modality. [12]

Fagel [12] claims that the stimulus with conflicting audio and visual content can be perceived as an emotion which is neither the emotion indicated by the audio information nor the emotion indicated by the visual information, which is called emotional McGurk Effect. It is assumed that the valence (positive or negative emotion) is primarily conveyed by the visual channel while the degree of arousal is reflected by the audio channel. A match of a positive facial expression with a negative voice will be perceived as joy. However, the identification of content emotion will be derived from the combination of sad voice with happy facial expression. Other mismatches between audio and visual information are only perceived as either the emotion indicated by audio channel or the emotion indicated by visual channel.

Since the encoding and decoding of emotion may depend on multiple modalities and language backgrounds, the purpose of the present study is to clarify the process of the encoding and decoding of emotion through a cross-cultural perceptual experiment for multi-modal emotions. The preliminary analysis on Chinese and Japanese listening to emotional speech of a Chinese speaker in three conditions of congruent audio-video, audio-only and video-only, reveals that language and culture will impose an influence on the identification of emotion. In the present paper, we will continue to explore the emotional speech communication but modulated in conflicting AV channels. Here, the issues concerned are as: (i) what is the interplay between the two conflicting AV channels in conveying emotional information? (ii) Does the emotional McGurk effect exist when the emotions are conveyed in conflicting channels? (iii) Are there any culture effects on perception on the conflicting AV emotions?

The assumptions are: (i) When listeners decode the conflicting AV stimuli, they might rely on some modality more than others across different emotions, i.e. one modality should have stronger emotional modulation for some emotions than that in another modality. (ii) Although the common psychological factor contributes to the emotional communication, the decoding of conflicting AV information will be affected by linguistic and cultural background and (iii) the emotional McGurk effect may also be related to culture norms of the encoder/listener.

2 Perceptual Experiment on Cross-Cultural Emotion

Table 1 lists the Chinese and the corresponding Japanese prompts. In order to control the time spent in the experiment, the prompts were divided into two sets. The sentences

were matched in the number of syllables, from 1 to 5, with different tonal combinations, in different grammatical structures. The contents of the texts were emotionally neutral.

The speech data used in the present paper is from a Chinese female student from Beijing Film Academy, who speaks Standard Chinese. Her emotion speech was videotaped with Canon Power Shot TX1 in the sound-proof room. She uttered the prompts in Table 1 in seven emotional states. The seven emotions are classified by valence (positive or negative emotions) : Happiness is positive and ‘Sadness, Anger, Disgust and Fear’ are positive; while ‘Surprise’ can be positive or negative. In terms of the degree of arousal, ‘Sadness and Fear’ are being low arousal, while ‘Happiness, Anger and Disgust’ are being high arousal.

Table 1. Chinese and Japanese prompt

Set 1	Set 2
S1-1 妈 お母さん (mother)	S1-4 骂 ののしる (to blame)
S2-1 大妈 お婆さん (auntie)	S2-2 踢球 サッカーをする (to play football)
S3-1 吃拉面 ラーメンを食べる (to eat noodle)	S3-2 奥运会 オリンピック (Olympic Games)
S4-1 打高尔夫 ゴルフをする (to play golf)	S4-2 足球比赛 サッカーの試合 (football match)
S5-1 张雨吃拉面 張雨さんはラーメンを食べる (Zhangyu eats noodles.)	S5-2 滑雪场教练 スキー場のスキーコーチ (coach of ski resort)

In order to explore the conflicting channel and the McGurk phenomenon, conflicting AV stimuli were obtained through dubbing a visual emotion with another vocal emotion for the same sentence. Then $5*7*7=245$ dubbed stimuli were obtained for each set including 35 congruent AV tokens.

Listeners were 10 Chinese college students not knowing Japanese and 10 Japanese college students not knowing Chinese. They were recruited to identify the emotional states for all the dubbed stimuli and rate the expressive degrees on a 5-point scale (0-4), multiple choices are allowed. The higher the score the more expressive the stimulus is.

3 Results and Analysis

The perceived scores were averaged for each intended emotions to obtain the confusing pattern for Chinese and Japanese listeners respectively. To depict the perceptual patterns clearly, a kind of spider graphs representing the average perceptual scores from these 10 Chinese and 10 Japanese listeners are plotted in Figures 1 to 4. Each ring in the graph represents the distribution of the rating scores of one perceived emotion for combinations of one facial (/vocal) expression (modality 1) and seven vocal (/facial) expressions (modality two). These rings are called here *Emotion Rings*. The changes in the shape and radius reflect the change of the perceptual patterns. If the

ring symmetrically distributes in all direction like a circle, then the perceived emotion is not related to the second modality. However, if the ring is in an unsymmetrical distribution, it means that the facial-vocal combination with higher scoring has a stronger tendency to be perceived as that emotion, on the contrary, the lower score the smaller chance. The variation of diameter size correlates with rating scores in various facial-vocal combinations.

3.1 Comparison on Perceptual Patterns

3.3.1 Perceptual Results for Chinese Listeners

(1) Fig. 1 (A) indicates that when 'Neutral' facial expression is dubbed with the seven vocal emotions, the two primarily perceived emotions are 'Neutral' and 'Surprise'. And the distribution patterns of the two emotion rings show a tendency to complement each other. The combinations of 'Neutral' face with 'Neutral', 'Happy', 'Fear', 'Sad' and 'Disgust' voices tend to be perceived as 'Neutral' while the combinations of 'Neutral' face with 'Angry' and 'Surprise' voices tend to be perceived as 'Surprise'. Fig.2 (A) shows that the combinations of 'Neutral' voice with varied facial expressions are mainly perceived as 'Neutral'. Except for the combination of a 'Neutral' voice with a 'Happy' face, which is perceived as 'Happy', almost all combinations are perceived as 'Neutral'. The combination of 'Neutral' voice with 'Angry' face is perceived as either 'Disgust' or 'Neutral' in equal probability, which is another exception.

(2) Fig.1 (B) indicates that the combinations of 'Happy' facial expression with varied emotional voices tend to be perceived as 'Happy', which is illustrated by an evenly distributed emotion ring. It means that the perception of 'Happy' depends more on visual information than audio information, although the facially 'Happy' emotion also initiates the percept of 'Surprise' and 'Neutral' as shown by the two small rings in the center. Fig.2 (B) reveals that the combinations of 'Happy' voice with varied facial expressions (except for 'Happy' face) could not be correctly perceived as 'Happy'. The combinations of 'Happy' voice with 'Sad', 'Surprise', or 'Neutral' faces are perceived as 'Neutral' emotion.

(3) Fig.1 (C) displays the complicated perceptual patterns activated by dubbing 'Angry' face with varied emotional voices. The integrations of 'Angry' face with 'Happy', 'Disgust' and 'Angry' voices can be perceived as 'Anger', 'Surprise' or 'Disgust' with almost equal scores. Fig.2 (C) shows that the combinations of 'Angry' voice with varied facial expressions are primarily perceived as 'Surprise'. Only the combination of 'Angry' voice with 'Happy' face is perceived as 'Happy'.

(4) Fig.1 (D) shows that the combinations of 'Disgust' face with varied emotional voices could not be correctly perceived as 'Disgust'. When 'Disgust' face goes with 'Disgust', 'Surprise', 'Angry' and 'Happy' voice, the percept of 'Surprise' is induced. When 'Disgust' facial expression is combined with 'Neutral' voice, the percept of 'Neutral' emotion is initiated. The combination of 'Disgust' facial expression with 'Sad' voice is perceived as 'Sad' with very low rating scores. Fig.2 (D) specifies that

the perceptual pattern of combinations of '*Disgust*' voice with varied facial expressions is similar to that shown in Fig.1 (D), with most combinations being perceived as '*Surprise*' except that the combination of '*Disgust*' voice with '*Happy*' facial expression is perceived as '*Happy*' and the combination of '*Disgust*' voice with '*Neutral*' face is perceived as '*Neutral*'.

(5) Fig.1 (E) reveals that most combinations of '*Fear*' expression with varied emotional voices could not be correctly perceived as '*Fear*'; instead, two obvious rings of '*Surprise*' and '*Neutral*' emotion are displayed. The percept of '*Surprise*' is induced when '*Fear*' expression is combined with '*Disgusted*', '*Surprise*' or '*Angry*' voices. The percept of '*Neutral*' emotion is initiated when '*Fear*' expression is dubbed with '*Neutral*' voice. However, the result is vague when '*Fear*' expression is dubbed with '*Sad*', '*Happy*' or '*Fear*' voices. Fig.2 (E) shows that the perception of '*Fear*' voice with varied facial expressions is ambiguous with rating scores lower than two points except that the combination of '*Fear*' voice with '*Happy*' face brings a percept of '*Happy*'.

(6) Fig.1 (F) shows that when '*Sad*' face is dubbed with varied emotional voices, two emotion rings of '*Surprise*' and '*Neutral*' are exhibited in a symmetrical pattern. Specifically, the combinations of '*Sad*' face with '*Angry*', '*Surprise*' and '*Disgust*' voices lead to the percept of '*Surprise*' and the combinations of '*Sad*' face with '*Neutral*' and '*Happy*' voices are perceived as '*Neutral*' emotion. When '*Sad*' face goes along with '*Fear*' or '*Sad*' voice, either '*Neutral*' or '*Sad*' is perceived with almost equal scores. Fig.2 (F) reveals that two overlapped emotion rings are formed when '*Sad*' voice is combined with varied facial expressions (except for '*Happy*' face), namely, '*Sad*' and '*Neutral*' emotion rings. However, their rating scores are very low, which are less than 2 points. The combination of '*Sad*' voice with '*Happy*' face is more likely to be perceived as '*Happy*'.

(7) The two symmetrically distributed emotion rings in Fig.1 (G) display the patterns of the '*Surprise*' face dubbed with varied emotional voices: a '*Surprise*' ring derived from the combinations of '*Surprise*' face with '*Angry*', '*Surprise*' and '*Disgust*' voices; and a '*Neutral*' emotion ring derived from the combinations of '*Surprise*' face with '*Neutral*', '*Happy*', '*Sad*' and '*Fear*' voices.

In Fig.2 (G), the perceptual pattern of *surprised* voice with varied facial expressions is represented by a dominant '*Surprise*' emotion ring. But the combination of '*Surprise*' voice with '*Happy*' face is inclined to be perceived as '*Happy*'.

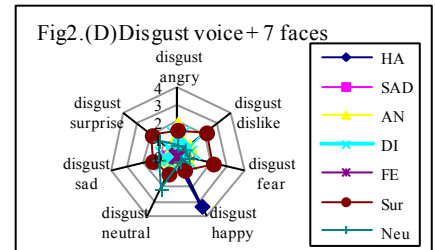
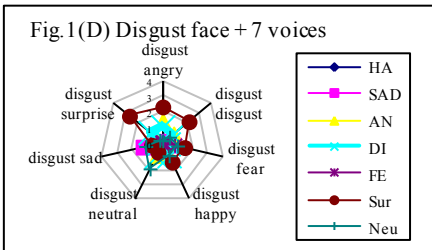
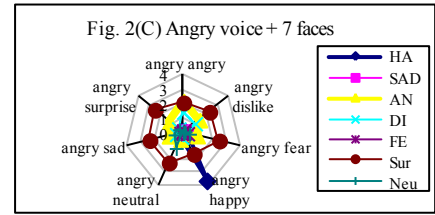
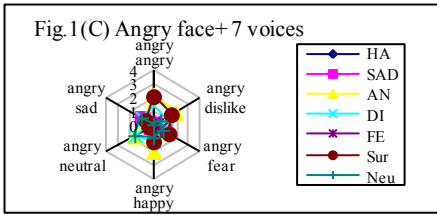
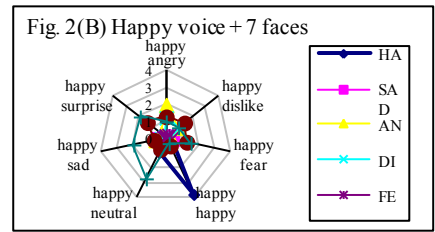
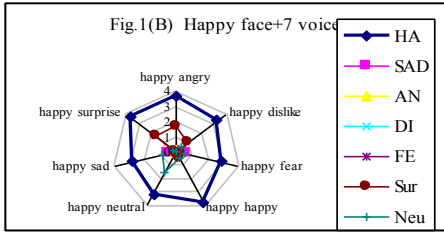
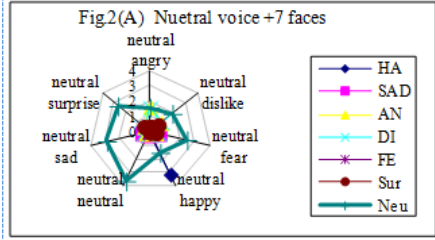
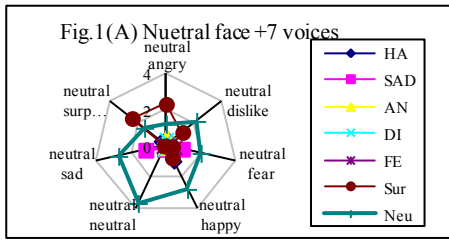


Fig. 1. Graphs from (A) to (G) reveal the perception patterns for the combinations of each facial expression with seven vocal expressions(10 Chinese listeners)

Fig. 2. Graphs from (A) to (G) reveal the perception patterns for the combinations of each emotional voice with seven facial expressions (10 Chinese listeners)

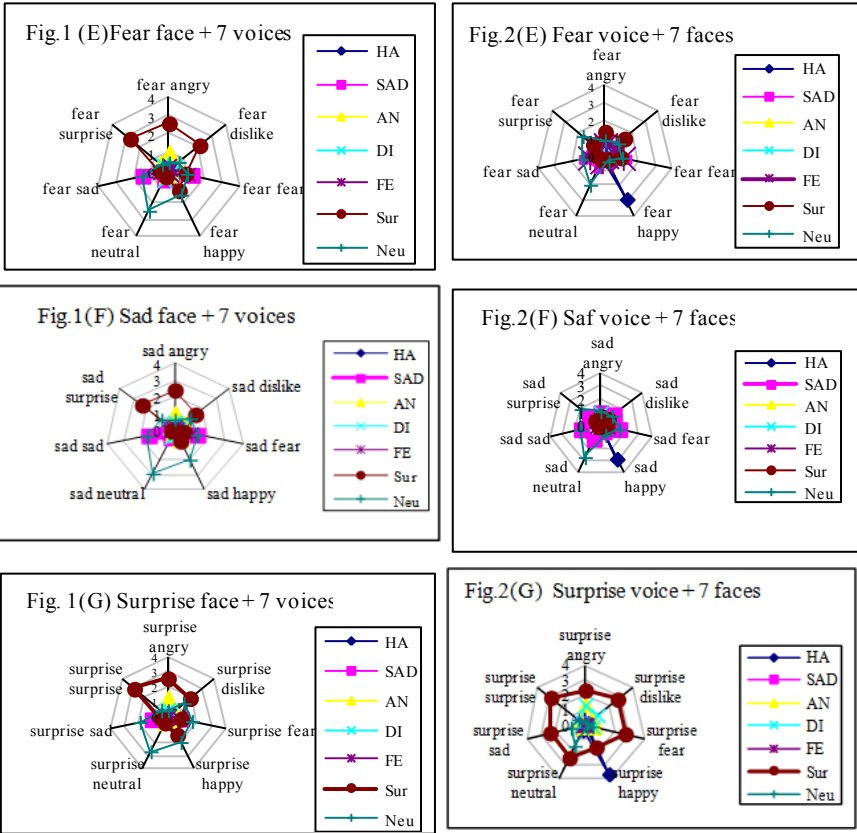


Fig. 1. (continued)

Fig. 2. (continued)

3.1.2 Perceptual Results for Japanese Listeners

(1) Fig.3 (A) indicates that when ‘Neutral’ face is dubbed with non-‘Neutral’ voices, the emotion stimuli are primarily perceived as ‘Neutral’. Fig.4 (A) shows that the combinations of ‘Neutral’ voice with varied facial expressions could not lead to the dominance of any emotion ring, except that the combination of ‘Neutral’ voice with ‘Happy’ face is regarded as ‘Happy’; and neutral voice with ‘Neutral’ face is regarded as neutral. The combination of ‘Neutral’ voice with an ‘Angry’ or ‘Disgust’ face is perceived as either ‘Disgust’ or ‘Anger’, with almost equal scores less than 2 points.

(2) In Fig.3 (B), the large and symmetrically distributed emotion ring shows that the combinations of ‘Happy’ face with varied emotional voices are perceived as ‘Happy’, signifying that the visual modality contributes more than the audio modality in the identification of ‘Happy’. In other words, the perception of ‘Happy’ can be independent of the audio modality. Fig.4 (B) reveals that the combinations of ‘Happy’

voice with varied facial expressions (except for 'Happy' face) could not be perceived as 'Happy'. The combinations of 'Happy' voice with 'Angry' or 'Disgust' faces are most likely to be perceived as 'Angry' and then as 'Disgust'. The combination of 'Happy' voice with 'Neutral' face is mainly perceived as 'Neutral'.

(3) Fig.3 (C) presents two dominant rings, with the 'Disgust' emotion ring embedded in the 'Angry' emotion ring. This perceptual pattern is triggered by integrating 'Angry' facial expression with varied emotional voices. Fig.4 (C) shows that the integrations of 'Angry' voice with varied facial expressions are primarily perceived as 'Angry'. Only the integration of 'Angry' voice with 'Happy' facial expression is perceived as 'Happy'.

(4) Fig.3 (D) presents two similar evenly distributed emotion rings: 'Disgust' ring and 'Angry' ring, which are resulted from the combinations of 'Disgust' facial expression with varied emotional voices. Fear can be perceived when 'Disgust' facial expression is dubbed with 'Fear' or 'Sad' voice. Fig.4 (D) specifies that the combinations of 'Disgust' voice with 'Angry', 'Happy' and 'Neutral' facial expressions are perceived as 'Angry', 'Happy' and 'Neutral' respectively, while the perceptual scores of other combinations are very low and show no obvious tendencies.

(5) Fig.3 (E) shows no obvious perceptual tendency for the combinations of 'Fear' face dubbed with varied emotional voices (scores < 2 points). Fig.4 (E) reveals that except that the perception of the combination of 'Fear' voice with 'Happy' face is identified as 'Happy' and the combination of 'Fear' voice with 'Neutral' face is identified as 'Neutral', all the scores of the combinations of 'Fear' voice with other facial expressions are lower than 2 points.

(6) Fig.3 (F) shows no obvious perceptual tendency under conditions of 'Sad' facial expression with varied emotional voices. Fig.4 (F) also reveals that there is no obvious perceptual tendency (scores < 2 points) when 'Fear' voice is combined with varied facial expressions except for 'Happy' and 'Neutral' facial expressions. The perception scores for 'Fear' and 'Sad' are equal. The combinations of 'Fear' voice with 'Happy' and 'Neutral' facial expressions tend to be perceived as the emotion implied in the facial expression.

(7) Fig.3 (G) shows no obvious perceptual tendencies under conditions of 'Surprise' face with varied emotional voices. Each perceptual score is less than 2 points. It can be concluded from Fig.4 (G) that the combinations of surprised voice with varied facial expressions cannot be recognized as 'Surprise'; the combination of surprised voice with 'Happy' facial expression is inclined to be perceived as 'Happy'; the combinations of surprised voice with 'Angry' and 'Disgust' facial expressions tend to be perceived as 'Angry'; and the combination of surprised voice with 'Neutral' facial expression is perceived as 'Neutral'.

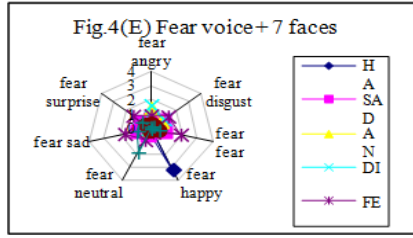
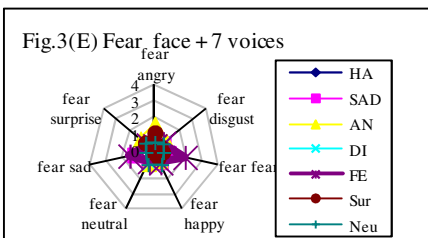
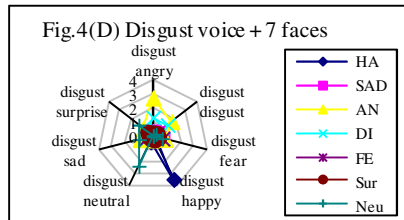
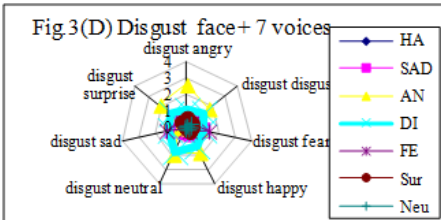
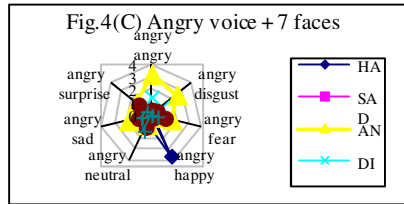
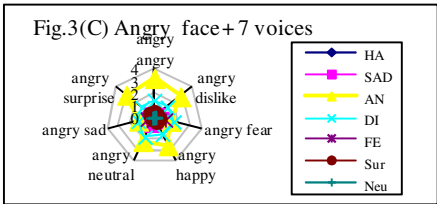
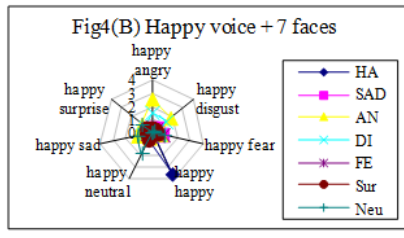
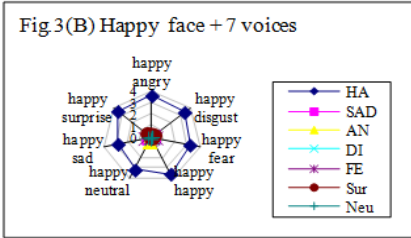
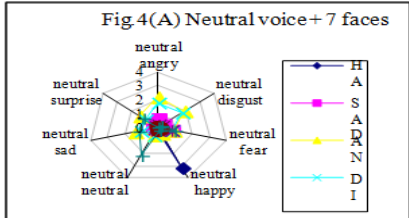
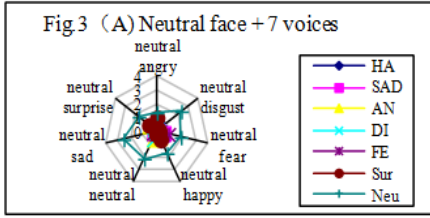


Fig. 3. Graphs from (A) to (G) reveal the perception modes for the combinations of each facial expression with seven voices under the AV-congruent and AV-conflicting condition (10 Japanese listeners)

Fig. 4. Graphs from (A) to (G) reveal the perception modes for the combinations of each emotional voice with seven facial expressions under the AV-congruent and AV-conflicting condition (10 Japanese listeners)

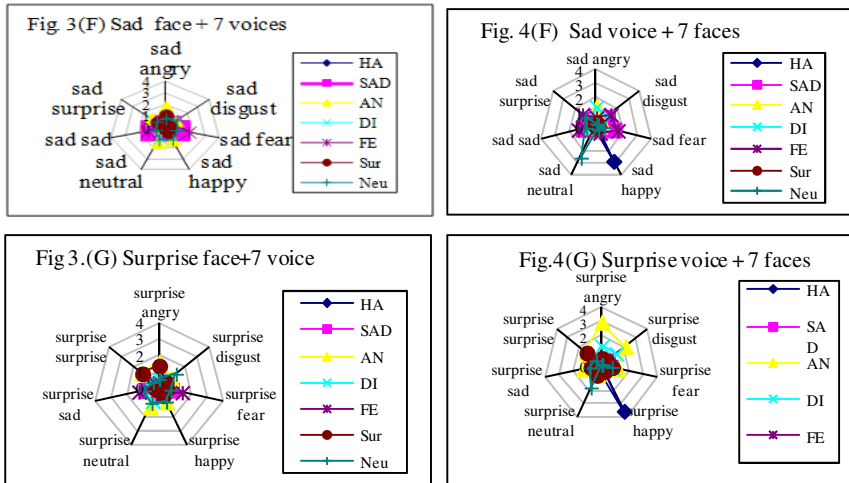


Fig. 3. (continued)

Fig. 4. (continued)

3.2 Comparison of Perceptual Patterns between Chinese and Japanese for Conflicting Stimuli

Figures 5~8 show the average perceptual score as a function of the intended emotion by vocal and facial expressions for Chinese and Japanese listeners in AV and CAV conditions. From the perspective of psychological dimension of emotion, Chinese and Japanese have similar perceptual patterns: for emotions with high arousal, the visual modality makes a major contribution to emotion decoding; while for emotions with low arousal, the audio modality makes a major contribution. In the AV-congruent setting, the perceptual scores of the Chinese for 'Neutral', 'Happy', and 'surprise' are higher than those of the Japanese, signifying higher confidence for the Chinese; while the scores for 'Angry', 'Disgust', 'Sad' and 'Fear' are lower than those of the Japanese, signifying lower confidence for the Chinese. The comparison of Fig. 5 with Fig. 7 indicates that there is a sharper drop in the scores of the Japanese than the Chinese according to vocal emotions. Fig. 6 and Fig. 8 reveal that the degree of falling according to facial emotions between the Japanese and the Chinese is similar except 'Surprise' and 'Neutral' emotion. The results may imply that, for Japanese listeners, decoding Chinese emotion counts more on the visual modality than the audio modality, and their decoding for 'Neutral' and 'Surprise' facial expression is better than the Chinese. It confirms the results in the previous study that in cross-cultural communication the facial information could help non-native listeners in emotion decoding than only vocal information. By further comparing Fig.5 with Fig.7, Fig. 6 with Fig.8, we find the tendency that Japanese listeners are consistent with Chinese listeners where visual modality exists, while they are discrepant for vocal modality conditions. The result supports the assumption that cross-cultural effect also exists when decoding information transmitted in incongruent channels, and that this effect is greater in the vocal channel than the facial channel.

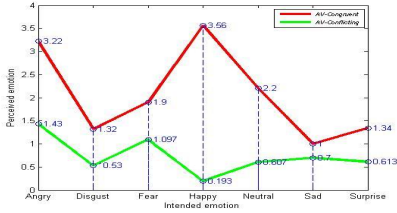


Fig. 5. Average perceptual score as a function of the intended emotion by vocal expressions for Japanese listeners in AV and CAV conditions

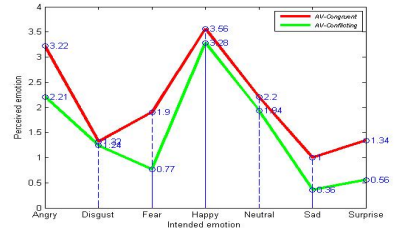


Fig. 6. Average perceptual score as a function of the intended emotion by facial expressions for Japanese listeners in AV-congruent and CAV conditions

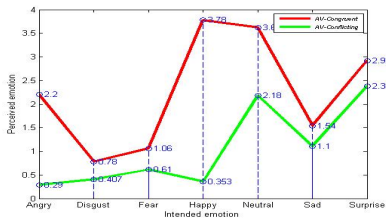


Fig. 7. Average perceptual score as a function of the intended emotion by vocal expressions for Chinese listeners in AV and CAV conditions

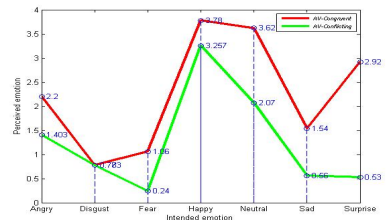


Fig. 8. Average perceptual score as a function of the intended emotion by facial expressions for Chinese listeners in AV and CAV conditions








3.3 Emotional McGurk Effect

Table 2 shows the cases of Emotional McGurk effect obtained for those cases with rating score > 2. For an instance, ‘Angry face + Disgust voice -> Surprise’ for Chinese listener; ‘Happy voice + Angry face -> Disgust’ for Japanese listener.

It was shown that Emotional McGurk effect distributes differently between Chinese and Japanese. There is only one set of combination (surprise voice with various facial expressions) where McGurk effect is not observed. But there are four sets of combinations (neutral, fear, happy facial expression with various voice and angry voice with various facial expressions) where McGurk effect is not observed. The “third emotion” in McGurk effect is most likely to be surprise and more likely to be neutral for Chinese but most likely to be anger and more likely to be disgust for Japanese. These results demonstrate that culture has effect on the decoding of emotion. Taking a further look at the Chinese data, we find that the case where McGurk effect is observed is normally related to either a negative vocal or a negative visual expression (here ‘Disgust, Sad and Surprise’ emotions as negative emotions. In fact, ‘Surprise’ is an ambiguous emotion state concerning intended negative or positive one, here it is expressed more negative). Visual expression of ‘Surprise, Fear and Sadness’ tend to be perceived as ‘Neutral’ emotion. For Japanese, the combination where McGurk effect

is observed is also the one where either vocal expression or visual expression of emotion is negative. Of these combinations, as long as visual modality indicates the emotion of anger, the combination will be recognized as ‘Disgust’, most of the other cases will be decoded as ‘Angry’.

Table 2. Emotional McGurk effect containing only the scores marked by 2 or 3 asterisks

Facial	Chinese		Japanese		Vocal	Chinese		Japanese	
	Vocal	Perceived	Vocal	perceived		Facial	perceived	Vocal	perceived
Ne 	AN	SU***	----	----	Ne	AN	DI**	AN DI/SA/SU	DI** AN**
AN 	Di Ne	SU** DI**	FE/HA/Neu/SA	DI**	AN	F/Neu/S DI HA	SU*** SU*** SU**	----	----
FE 	An DI HA/SA	SU*** SU** Neu**	---	---	FE	SU	Neu**	AN	DI**
HA 	AN	SU**	----	----	HA	FE/SA/SU	Neu**	DI AN	AN** DI**
DI 	AN	SU***	HA/Neu SU	AN** AN***	DI	AN /FE/SA	SU**	SU	Ne**
SA 	AN DI HA	SU*** SU** Neu**	Ne/SU	AN**	SA	FE/SU	Neu**	AN	DI**
SU 	FE/HA/SA	Neu**	DI Ne	Ne** AN**	SU	--	--	DI SA	AN*** AN**

4 Conclusions

The main conclusion is that cultural background poses difference in the perceptual patterns between the Chinese and the Japanese. From the perspective of psychological dimension of emotion, in the AV-conflicting setting, the Chinese and the Japanese have similar patterns: for emotions with high arousal, the visual modality makes a major contribution to emotion decoding; while for emotions with low arousal, the audio modality makes a major contribution. Due to linguistic and cultural difference, the Chinese listeners make more use of the audio modality to decode emotion; for those Japanese who don’t know Chinese, the emotion recognition counts more on the visual modality and their decoding of ‘Neutral’ and ‘Surprise’ facial expression is better than that of the Chinese. Regarding to the rating confidence, the Chinese give higher scores than the Japanese. The findings here are different from those in [12], which assumed that the visual modality mainly transmits valence (positive or negative emotion) and the audio modality mainly transmits arousal (the degree of excitement). One explanation for this discrepancy lies in the difference in the number of emotions: seven in our research and only four in [12].

The emotional McGurk effect is found in the AV-conflicting experiment. Though the occurrence of the McGurk effect relates highly to negative emotions, the perception patterns are different due to the culture effect. Inconsistent with Fagel [56], no

cases of the emotional McGurk effect relating to positive emotions are found. To some extent, the occurrence frequency of the McGurk effect shows that the Chinese listeners have a tendency to jump to a conclusion ('*Surprise*') while the Japanese favor ambiguity ('*Anger*', '*Disgust*' or '*Neutral*').

The results support the assumptions that (1) When listeners decoding the conflicting AV stimuli, they might rely on some modality more than another across different emotions, as shown in Table 3-7. (2) Although common psychological factor contributes to the emotional communication, the decoding of conflicting AV information will be affected by culture background, and (3) the emotional McGurk effect exists, and it may also be related to cultural norms of the encoder/listener.

Future research will focus on more speakers from various cultures to verify the emotional McGurk effect patterns.

This work was supported by the National Basic Research Program (973Program) of China (No. 2013CB329301), NSFC Project with No. 60975081 and CASS innovation project.

References

1. Darwin, C.: The expression of the emotions in man and animals. John Murray, Oxford University Press, London, New York (1988, Original work published in 1872) (reprinted with introduction, afterword, and commentary by Ekman, P. (ed.))
2. Ekman, P., Friesen, W.V.: Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology* 17, 124–129 (1971)
3. Izard, C.E.: Innate and universal facial expressions: Evidence from developmental and cross-cultural research. *Psychological Bulletin* 115, 288–299 (1994)
4. Scherer, K.R.: A Cross-Cultural Investigation of Emotion Inferences from Voice and Speech: Implications for Speech Technology. In: *ICSLP 2000* (2000)
5. Scherer, K.R., Banse, R., Wallbott, H.G.: Emotion Inferences from Vocal Expression Correlate Across Languages and Cultures. *Journal of Cross-Cultural Psychology* 32, 76 (2001)
6. Scherer, K.R.: Vocal communication of emotion: A review of research paradigms. *Speech Communication* 40, 227–256 (2003)
7. Abelin, A., Allwood, J.: Cross Linguistic Interpretation of Emotional Prosody. In: *Proc. ISCA Workshop on Speech and Emotion, Belfast* (2000)
8. Yanushevskaya, I., Chasaide, A.N., Gobl, C.: Cross-Language Study of Vocal Correlates of Affective States. In: *Interspeech 2008* (2008)
9. Abelin, Å.: Cross-Cultural Multimodal Interpretation of Emotional Expressions – An Experimental Study of Spanish and Swedish. In: *SP 2004* (2004)
10. Huang, C.F., Akagi, M.: A three-layered model for expressive speech perception. *Speech Communication* 50, 810–828 (2008)
11. Barkhuysen, P., Kraemer, E., Swerts, M.: Incremental perception of acted and real emotional speech. In: *ICPHS* (2007)
12. Fagel, S.: Emotional McGurk effect. In: *Speech Prosody 2006* (2006)
13. Grandjean, D., Scherer, K.R.: Unpacking the Cognitive Architecture of Emotion Processes. *Emotion* 8(3), 341–351 (2008)
14. Erickson, D.: Expressive speech: Production, perception and application to speech synthesis. *Japan Acoust. Sci. & Tech.* 26, 4 (2005)

Pests Hidden in Your Fans: An Effective Approach for Opinion Leader Discovery

Binyang Li, Kam-fai Wong, Lanjun Zhou, Zhongyu Wei, and Jun Xu

Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong

Key Laboratory of High Confidence Software Technologies
Ministry of Education, China

{byli, kfwong, ljzhou, zywei, jxu}@se.cuhk.edu.hk

Abstract. With the development of Web 2.0, people would like to share opinions on the Web, which are very helpful for other users to make decisions. Especially, some users have more powerful influence to other members of a community, group, or society, and their advice, opinions, and views are more valuable. We call these people opinion leaders. The study of opinion leader discovery from the social media is meaningful because it could help users to understand influential user behavior, and trace vital information diffusion of an e-society, even on-line ecology. However, existing approaches focus on linkage-based methods without considering the *pests* who have relationship with the potential opinion leader but carrying opposite opinions. In an extreme case, an opinion leader might be mistakenly identified according to his richer relationships with the *pests*. In this paper, we start from explaining the definition of opinion leader, and take into consideration of the user profile and posts' opinions instead of using structural information (linkage) only. As such, those *pests* carrying opposite opinions could be gotten rid of from the social network, which could further improve the effectiveness of discovering opinion leaders. To evaluate the performance of our approach, we made experiments based on the Tweets data, and the results showed that our proposed approach could achieve 8% improvement compared with the linkage-based approach.

Keywords: Opinion leader, social network, Web 2.0, pests.

1 Introduction

With the development of Web 2.0, people would like to share opinions on the Web, which are very helpful for other users to make decision. Some of the users could always provide more valuable advice, and they gradually have more powerful influence to other members in the specific e-community, and further to the whole e-society. For example, *Gangnam Style* unusually drew the attention from the world in the past few months. Till November 2012, its MV uploaded in *Youtube* has been watched almost

8.4 hundred million times. How did *Gangnam Style* turn into a success? To trace backward, we found that opinion leaders gave most of the effort on this. At first, the click-through rate was poor. But then, some artists who have strong calling power, like *Britney Spears*, *Katy Perry*, promoted it keenly in Twitter. Influenced by them, artists from other areas shared it, as a result, the trend started to spread among the whole world. *Gangnam Style* became famous incredibly. It is clear that the opinion leader is the key of success.

The formal definition of opinion leader is that people who are influential members of a community, group, or society to whom others turn for advice, opinions, and views. A user is considered as an opinion leader when he involves the following factors¹:

- Expression of values
- Professional competence
- Nature of his social network

Due to the commercial factors of opinion leader, the study of opinion leader focuses on mining the commercial value of a given opinion leader, such as product suggestion, advertising campaign, and so on, while ignore how to discover opinion leaders. According to our preliminary study, there exists some related work on opinion leader identification, e.g. Central Policy Unit.[1] Unfortunately, most of the opinion leader analysis utilizes manual methods for discovering opinion leaders, rather than carrying out automatically. Manual methods may be work in Web 1.0 era with limited portal websites, but hardly continue when entering big data.[2] Besides, the research on automatically opinion leader discovery takes only the nature of the social network into consideration, e.g. PageRank, HITS like models.[10][11] However, not all the relationships between users express supporting opinions. For instance, “*Sister Phoenix*”(凤姐) has a great amount of fans and relations on *Weibo*, but most of her fans oppose her. It is very often that the relationship between an opinion leader and a non-opinion leader carries the opposite opinion. Therefore, we argue that distinguish pests from your fans or replies will improve the performance of opinion leader identification.

In this paper, we target to identify the opinion leaders on social media. We discover an opinion leader by accounting for all the three factors in the definition of opinion leaders. We present a 2-step approach for discovering opinion leaders: we first generate a group of candidate/potential opinion leaders according to the expression of values and professional competence by analyzing the profile and the post content of the users. Specifically, we propose two categories of features to (1) describe the user profile which will help detect the professional area and measure his/her competence; (2) analyze the content of his/her posts together with the replies and the comments which will help determine the value of the expression, e.g., get rid of the pests expressing opposite opinions to refine the social network. We then integrate the potential opinion leaders into a graph-based model to measure its nature of social networks.

¹ http://en.wikipedia.org/wiki/Opinion_leader

In order to investigate the performance of our proposed method, we also conduct several experiments based on the real data, which was collected from Twitter about the UK General Election in 2010[3]. We investigate the contribution of different features, including content-based, user-based and linkage-based for identifying opinion leaders. A comparative experiment is conducted and the experimental results showed that our proposed approach outperform the state-of-the-art linkage based method.

The rest of this paper is organized as follows: we will review the related works in Section 2. In Section 3, we will present our 2-step approach for opinion leader discovery. We evaluate our approach in Section 4. Finally, we will conduct the conclusion and suggest future works in Section 5.

2 Related Work

Most of the previously work about opinion leader focused on how to utilize its commercial value, such as marketing research, product sampling, retailing/personal selling, and advertising.[4][5][6] For this kind of research, the opinion leaders were pre-determined manually, and the data was not open to public.

Until the recent decade, with the explosion of information, automatically discovering opinion leader attracted more and more attentions.

Thomas, et al., proposed analytic hierarchy process (AHP) method, which is a structured technique for organizing and analyzing complex decisions.[7] According to each shortlisted user, assorted factors should be considered. Thus, the AHP provides a comprehensive and rational framework for quantifying its factors in order to relate those factors and evaluate the solutions, which mean the choices of opinion leaders in this case. An AHP hierarchy consists of an overall goal, a group of alternatives for reaching the goal, and a group of related factors. The factors can be further broken down into many levels as the problem requires.

Laclavik, et al., proposed a method for opinion leader identification based on the relationship network.[8] They suggested to determine the communication relationships between users by relationship mining methods. The extracted users and its relationship network formed a social network could be then represented as graphs. The resulting graph was analyzed by determining key figures for the position of single users and for the overall structure of the network. In this way, opinion leaders could be identified.

Centrality analysis approach was proposed to measure the degree of activist's connection with others.[9] The more the connections were, the more the influences of that activist on others. Opinion leaders could be listed out based on the degree of influence and activeness. Social network analysis provided a number of key features which described the structure of the entire network. For analyzing opinions, the key figures density, connectivity, and closeness centralization were especially relevant. Density measured the connection of a network and is an indicator for communication within the network.

In summary, all of the above approaches only focus on analyzing the relationship between the users. They make use of the concept of tree structure or graph structure to illustrate the influence brought by the opinion leaders. They concern the levels, width and size which can show the structural information of opinion leaders. However, structure-based methods only show the number of people replied or retweeted his or her posts but not going to consider the content of the tweets, i.e. opinions. This kind of content of a tweet can illustrate whether it supports the idea of the previous posts or whether they are talking about the same topic. If we find that the repliers do not agree with the viewpoints of the previous tweet, then this tweet cannot be counted as an effect of influence in his social network. Therefore, in this paper, we will account for the content-based features for opinion leader discovery.

3 Methodology

In this section, we will present our 2-step approach for opinion leader discovery. It is intuitively that the best way of identify the *website leader* is based on the analysis of relationships of websites where the link between two websites only carry “supporting” meaning. However, for opinion leader discovery from social media, the link between two users may indicate an “opposite” meaning or “none”. In other words, the traditional meaning of the linkage cannot tell the whole story, and there exists pests hidden in the relations carry opposite opinions. Therefore, we suggest to identify the opinion leaders by considering “opinion” and propose a 2-step approach where a candidate opinion leader set is firstly generated by considering the factor of sentiment and then implemented into a graph-based model for final identification.

3.1 Potential Opinion Leader Generation

Recall that a user is considered as an opinion leader when he/she involves the following factors: expression of values, professional competence, and nature of its social network. We propose a number of features to describe the first two factors in this subsection and put them into SVM[12] to generate a candidate opinion leader set. Then we implement the results into graph-based model for analyzing the nature of its social network to discover opinion leaders.

- Professional Competence

Since our target is to discover opinion leader from social media, we take Twitter as an example for further description. We list some statistics related to the users’ profession from his/her profile. This category of features represents the nature of user’s professional competence shown in Table 1.

Table 1. The descriptions of user-based features

Name	Description
Tweet Count	If the post contains more comments, it means that the scope of influence by the post increases. Also, when one user’s posts are being commented for more times, it proves that the user might have a greater influence.
Follower Count	This simply counts the number of followers of each user in order to measure his credibility/authority.
Verified	Verification is currently used to establish authenticity of identities on Twitter, which could improve the authority of the user.
Retweet Count	The value of the user’s professional degree can be reflected by the count of the retweeting the post.
Reply Count	The more number of reply, the higher attention from the other users have paid on the topic.
Retweet time range	To investigate the time of validity of a tweet.

- Value of Expression

In order to measure the expression of values, we analyze the content of the post together with its comments/replies. Accordingly, the category of content-based features is proposed to help us find out the viewpoint and argument of a comment as shown in Table 2.

Table 2. The descriptions of content-based features

Name	Description
Pos. Ratio of Reply	The pos. ratio tells us the percentage of reply having the same attitude towards the same side is. The higher the ratio is, the more the people agree with his or her viewpoints.
Cons. Ratio of Reply	The cons. ratio tells us the percentage of reply having the opposite attitude towards the author’s side is. The higher the ratio is, the more the people disagree with his or her viewpoints.
Sentiment Degree	It calculates the strength of the sentiment words within the tweet or reply.
Words Count	How many words are included in a tweet?
Positive Words Count	How many positive words are included in a tweet, which could help us to understand how strong its attitude is expressed?
Negative Words Count	How many negative words are included in a tweet, which could help us to understand how strong its attitude is expressed?
Hashtag	The Hashtag is used to mark keywords or topics in a Tweet.

To understand more about the attitude of users towards their opinion leaders, the reply's content is important. Does the user support or oppose the viewpoint of the opinion leader? In our method, we utilize *Sentiwordnet*² to help us analyze the opinion of the tweet. *Sentiwordnet* is an open source which consists of 17,370 negative words and 18,157 positive words. There is a score assigned to each individual sentiment word to indicate the positive or the negative strength of the sentiment word.[16] (In this paper, the supporting opinion and opposite opinion are also referred as positive opinion and negative opinion, respectively.)

For simplicity, we just measure the opinionatedness in two naive ways: calculating the scores of the sentiment words appearing in the tweet; counting the number of sentiment words. Then we sum up the score or the sentiment word count of a tweet to indicate the attitude (positive or negative).

We then put both categories of features into *Supporting Vector Machine* (SVM) to generate candidate opinion leaders.

3.2 Opinion Leader Identification

After we generate the candidate opinion leaders, we then integrate them into a graph-based model to rank and generate the final opinion leaders.

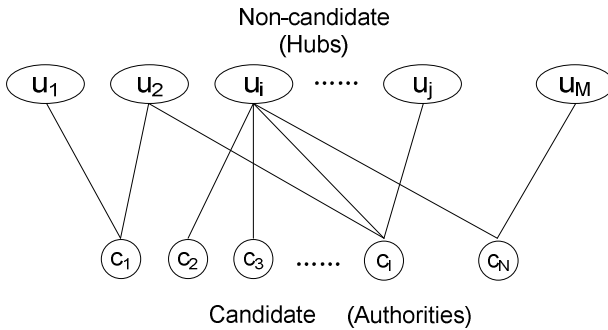


Fig. 1. Graph model for opinion leader identification.

Our proposed graph model is based on HITS algorithm, which distinguishes the users into hubs and authorities. An authority represents a candidate opinion leader, noted by c , while a hub indicates the non-candidate user, noted by u . For each individual u_i , it has links to many authorities. An authority c_j would have many hubs linking to it. The hub scores and authority scores are computed in an iterative way. Fig. 1 gives the graph model representation of the HITS model.

For our purpose, the non-candidate layer is considered as hubs and the candidate layer authorities. If a non-candidate user posts a reply or comment to support the opinion of candidate opinion leader, there will be an edge between them. In Fig. 1, we can see that the candidate opinion leader that has links from many non-candidate

² <http://sentiwordnet.isti.cnr.it/>

users can be assigned a high weight to denote a strong social network. On the contrary, if a candidate opinion leader has few links from the Hubs, the score is low, which will result in a low ranking. Each edge is associated with a weight w_{ij} denoting the contribution of u_i to the candidate opinion leader c_j . The weight w_{ij} is computed by the contribution of non-candidate users.

Different from existing approaches, we consider sentiment factor for opinion leader discovery. We divide the links between users into supporting (positive) ones and opposite (negative) ones, and regard those positive links are valuable in its social network. Therefore, we filter out the pests from follower with links expressing opposite opinions more than supporting ones in the first step. We then compute the weight of the edge by only accounting for positive links.

For computation of the final scores, the initial scores of all candidate opinion leaders are set to $1/N$, and non-candidate are set to $1/M$. The above iterative steps are then used to compute the new scores until convergence. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any nodes falls below a given threshold [13][14][17]. In our model, we use the authority scores as the total scores. The opinion leaders are then ranked based on the total scores.

4 Experiment

4.1 Experimental Setup

Dataset

The tweets data we used in this paper were collected using the Twitter Streaming API for 8 weeks leading to the UK General Election in 2010[3]. Search criteria specified include the mention of political parties such as Labour, Conservative, Tory1, etc., the mention of candidates such as Brown, Cameron, Clegg, etc., the use of the hash tags such as #election2010, #Labour etc., and the use of certain words such as election. The corpus contains around 919,662 unique tweets and 68,620 users. We also collected the following links of all the users.

According to the Twitter setting, data can only be revealed at most one-week quantity each time. Thus, data sample will be separated randomly into one-week sized in this case for afterward analysis in order to fit the circumstance, which data is extracted automatically in the future. Thus, the dataset for analysis contains around 44,391 unique tweets and 18,713 users.

Annotation

We have also done the clustering for the data before the annotation. There are 4 subgroups such as conservative party, labour party, liberal democrat party and others. For each subgroup, we have annotated the opinion leaders in the subgroup manually while two out of three members in our group agree that the user is opinion leader, which mean more than half of the group members agree, then the user is opinion leader. And it is the majority rule to identify the opinion leaders manually. The Kappa coefficient[18] indicating inter-annotator agreement was 0.8236 for the binary

classification. The conflict labels from the two annotators were resolved by a third annotator. Finally, there are 129 opinion leaders annotated in the training dataset.

In our experiment, the data is divided into five folds and four of them are training data and the one left is testing data. In order to investigate the performance of 2-step approach, we compare our proposed method with the linkage-based approach, which achieved the best run. Beside, we have proposed several categories of features to identify potential opinion leaders in Section 3, which are content-based and user-based mentioned above. In order to investigate the effectiveness of each category of features, we also tried different combination of feature sets.

Baseline

We choose linkage-based method as the baseline, which achieved best performance among linkage-based methods[15].

Metrics

We utilize *precision recall* and *f-value* as our evaluation metrics. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The recall is intuitively the ability of the classifier to find all the positive samples. The F-value can be interpreted as a weighted harmonic mean of the precision and recall. A measure reaches its best value at 1 and worst score at 0.³

4.2 Experimental Result

The overall performance of different approaches is shown in Table 3. We use SVM+C to denote Content-based feature, SVM+U means user-based feature, and SVM+ALL stands for adding them all.

The experimental results showed that the 2-step approach with our proposed features outperformed the baseline. Especially, when all the features were taken into consideration, the F-value was the highest which identified 106 opinion leaders and achieved around 8% improvement over the baseline. Moreover, there more than 10 fake opinion leaders identified by the baseline due to their well social network but mainly negative comments.

We further compared the features in different categories, the content-based features were more important than user-based features, which could identify 94 opinion leaders.

Table 3. Comparison between different approaches for opinion leader discovery

Approach	Precision	Recall	F-value
Baseline	0.7322	0.6285	0.6763
SVM+C	0.8032	0.6345	0.7089
SVM+U	0.7740	0.6316	0.6955
SVM+ALL	0.8180	0.6351	0.7150

³ http://scikit-learn.org/dev/modules/model_evaluation.html

We further investigate the performance of each individual feature of content-based category, and the results were shown in Table 4.

It is clear that SVM+ratio achieved the best run, which demonstrated that the feature of pos. or cons. ratio was the most effective way to decrease the impact of relationship carrying negative opinion comparing with the baseline. Besides, according to our analysis of the experimental results, one would like to express opinions to attack others during the Election rather than post replies for supporting. As a result, if a candidate opinion leader has high pos. ratio, it is probable to be an opinion leader.

Table 4. Comparison between different features for opinion leader discovery

Approach	Precision	Recall	F-value
SVM+Pos./Cons. ratio of reply	0.8089	0.6317	0.7094
SVM+Sentiment word count	0.7461	0.6234	0.6792
SVM+Sentiment degree	0.7475	0.6267	0.6817
SVM+Word count	0.7418	0.6182	0.6685
SVM+Hashtag	0.7122	0.6085	0.6563

5 Conclusion and Future Work

5.1 Conclusion

This paper targets to identify opinion leaders on the social media. The main difference from traditional *website leader* is that the link of social network doesn't always mean supporting. A pest link is likely to exist with negative opinions between users. We, therefore, design a 2-step model by taking into consideration of the key factors of opinion leaders, the value of expression, the professional competence, and the nature of social networks.

Specifically, a candidate opinion leader set is generated by utilizing the user profile to detect the professional area and measure user's competence; by analyzing the content of user's posts to determine the value of the expression in the first step. In the second step, a HITS-like graph is constructed based on the potential opinion leaders to rank the opinion leaders.

In conclusion,

1. We propose a set of useful features to describe the key factors of opinion leaders, which is proved to be effective for candidate opinion leader generation;
2. A graph-based model is devised for ranking opinion leaders, which decreases the impact of links with negative opinions;
3. A 2-step approach for opinion leader identification is presented from the perspective of view of opinion leader definition;
4. Several experiments were conducted and the results showed the effectiveness of our proposed 2-step approach, which could achieve 8% improvement over the baseline.

5.2 Future Work

In the future, we will continue our research on opinion leader discovery in the following directions:

1. Develop a unified model for opinion leader discovery by considering information diffusion;
2. Implement the fine-grained opinion analysis into content analysis, e.g. opinion target identification[19];
3. Classify the comments into different categories so as to build up the relationship between comments.
4. Besides Tweets, we would like to move forward to other data from different languages of social media, like Weibo, My Space, etc.

Acknowledgments. This work is partially supported by National 863 program of China (Grant No. 2009AA01Z150), General Research Fund of Hong Kong Research Grants Council (Project No. 417112), and CUHK Direct Grants (No. 2050525). We also thank Xu Han and anonymous reviewers for their helpful comments.

References

1. Lewis, J.: The Search for Coordination: The Case of the Central Policy Review Staff and Social Policy Planning, 1971–1977. *Social Policy & Administration* 45(7), 770–787 (2011)
2. Asur, S., Huberman, B.A.: Predicting the future with social media. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 492–499. IEEE (2010)
3. He, Y., Saif, H., Wei, Z., et al.: Quantising Opinions for Political Tweets Analysis. In: Proceeding of the Eighth International Conference on Language Resources and Evaluation (LREC) (2012) (in submission)
4. Wang, J.C., Chen, C.L.: An automated tool for managing interactions in virtual communities-using social network analysis approach. *Journal of Organizational Computing and Electronic Commerce* 14(1), 1–26 (2004)
5. van der Merwe, R., van Heerden, G.: Finding and utilizing opinion leaders: Social networks and the power of relationships. Division of Industrial Marketing, eCommerce and Supply Chain Management, Luleå University of Technology, Luleå, Sweden (2009)
6. <http://www.opinionleader.co.uk/>
7. Saaty, T.L., Peniwati, K.: *Group Decision Making: Drawing out and Reconciling Differences*. RWS Publications, Pittsburgh (2008) ISBN 978-1-888603-08-8
8. Zhang, X., Dong, D.: *Way of Identifying the Opinion Leaders in Virtual Communities* (July 2008)
9. Laclavík, M., Dlugolinský, Š., Šeleng, M., et al.: Email analysis and information extraction for enterprise benefit. *Computing and Informatics* 30(1), 57–87 (2012)
10. Page, L., Brin, S., Motwani, R., Winograd, T.: *The pagerank citation ranking: Bringing order to the web*. Technical report, Stanford University (1998)
11. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *J. ACM* 46(5), 604–632 (1999)

12. Cortes, C., Vapnik, V.: Support vector machine. *Machine Learning* 20(3), 273–297 (1995)
13. Li, B., Zhou, L., Feng, S., Wong, K.-F.: A Unified Graph Model for Sentence-based Opinion Retrieval. In: *Proceedings of ACL 2010* (2010)
14. Wan, X., Yang, J.: Multi-document summarization using cluster-based link analysis. In: *SIGIR 2008: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 299–306. ACM (2008)
15. Cho, Y., Hwang, J., Lee, D.: Identification of effective opinion leaders in the diffusion of technological innovation: A social network approach. *Technological Forecasting and Social Change* 79(1), 97–106 (2012)
16. Taboada, M., Brooke, J., Tofiloski, M., et al.: Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37(2), 267–307 (2011)
17. Erkan, G., Radev, D.R.: LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)* 22, 457–479 (2004)
18. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics* 22(2), 249–254 (1996)
19. Zhou, L., Xia, Y., Li, B., Wong, K.-F.: WIA-Opinmine System in NTCIR-8 MOAT Evaluation. In: *NTCIR-8 Workshop Meeting* (2010)

Exploiting Lexicalized Statistical Patterns in Chinese Linguistic Analysis

Yu Zhao and Maosong Sun

Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology,
Tsinghua University, Beijing 100084, China

Abstract. The web corpus has been used for linguistic analysis with the help of search engines. In this paper, we describe the concept of lexicalized patterns, which we exploit to obtain statistical information using the simple string matching strategy via search engines. We discuss the usage of lexicalized statistical patterns at three linguistic levels of Chinese analysis: lexical, syntactic and semantic. We develop a specialized search engine to get frequency counts for these patterns on SogouT¹ corpus. Experimental results show that lexicalized statistical patterns are effective on analyzing the cohesion of phrases, determining the phrasal category and discovering patient objects.

Keywords: Lexicalized statistical pattern, Chinese linguistic analysis, Web corpus, Natural language processing.

1 Introduction

Most of current statistical natural language processing systems rely on large well-organized annotated corpus. For example, the state-of-the-art dependency parser uses Treebanks to extract POS-tag features [9]. Nevertheless, these corpora are highly time-consuming and labor-intensive to build and extend. Moreover, they are mostly in limited size. The main cause of error for many natural language processing task is the lack of related statistical information in the training set.

Let us consider the task of determining the phrasal category, which is one of the most significant issues in shallow parsing. In Chinese, a chunk that has two components: VP+NP, can possibly be a verbal phrase or a substantive phrase. For instance, 告别仪式 (farewell ceremony) and 告别朋友 (say goodbye to a friend) are both composed by VP+NP. while the former is a substantive phrase and the latter is a verbal phrase. However, the famous Stanford parser incorrectly categorizes 告别仪式 as a verbal phrase, as shown in Figure 1. Resolving this type of error requires information that is not present in Treebanks.

Therefore, a growing number of researchers have been realizing the potential of web-scale corpus to NLP tasks. The key advantage of web corpus lies in

¹ The 2008 version is available online at <http://www.sogou.com/labs/dl/t.html>

```

(ROOT[. $. / . $$ . ] [55.694]
  (IP[哭/VV]
    (PP[在/P] (P[在/P] 在)
      (LCP[上/LC]
        (IP[告别/VV]
          (VP[告别/VV] (VV[告别/VV] 告别)
            (NP[仪式/NN] (NN[仪式/NN] 仪式))))
          (LC[上/LC] 上)))
        (NP[我们/PN] (PN[我们/PN] 我们))
        (VP[哭/VV] (VV[哭/VV] 哭) (AS[了/AS] 了))))

```

Fig. 1. An error occurred in the parse output of the Stanford parser. The correct tag of phrase “告别 仪式” (a farewell ceremony) should be NP instead of VP.

its massive scale, which contributes to the completeness of statistical information. The main challenge is that web corpus usually consists of huge amounts of unstructured plain-text documents. As far as we know, there is not a flexible approach for analyzing the deep structure of natural language automatically, so that the major methodology of utilizing web search engine is simply based on string-matching strategy.

In this paper, we propose to design valuable patterns for different NLP tasks and access word count statistics via the search engines. For example, we find that 告别了朋友 (have said goodbye to a friend) appears more frequently than 告别了仪式 (have said goodbye to a ceremony) by comparing the amount of results acquired from web search engine². Furthermore, it indicates that for VP NP phrases, the occurrence of “VP+了+NP” structure is probably a discriminative cue to determine its phrasal category.

We denote structures like VP+了+NP as **templates**. The instantiated cases for the given template are defined as **lexicalized patterns**, such as 告别了仪式 and 告别了朋友. Our work is based on a reasonable hypothesis that the phrase frequency information collected from large corpora can reflect the correctness of phrase usage. If a lexicalized pattern frequently occurs in web-scale corpora, we would be confident that it is linguistically valid. In contrast, if a lexicalized pattern hardly appears, we would say that it is linguistically invalid. We denote those lexicalized patterns with frequency counts as **lexicalized statistical patterns**, examples of which are shown in Table 1.

Analysis of the Chinese language can be performed at different linguistic levels. In this paper, we discuss the usage of lexicalized statistical patterns on three levels of Chinese grammatical analysis. At the lexicalization level, we focus on analyzing the cohesion of compound noun phrases. At the syntactical level, we focus on determining the phrasal category. At the semantic level, we focus on discovering patient objects among the predicate-object phrases. We integrate templates and utilize valuable lexicalized statistical patterns at each level.

² We obtain 745,000 results for 告别了朋友, and 197 results for 告别了仪式 on Google.

Table 1. Frequency information of three lexicalized statistical patterns at different linguistic levels. Here, 兔子的尾巴 means *the tail (尾巴) of rabbits (兔子)* in English. 把食堂吃了 means *to eat (吃) the canteen (食堂)* in English while the original phrase 吃食堂 means *eating at the canteen*.

	Lexical level	Syntactic level	Semantic level
Phrase	兔子(NP ₁) 尾巴(NP ₂)	告别(VP) 朋友(NP)	吃(VP) 食堂(NP)
Template	NP ₁ +的+NP ₂	VP+了+NP	把+NP+VP+了
Pattern	兔子的尾巴	告别了朋友	把食堂吃了
Frequency	272	8460	0

Recent works have shown that exploiting web-scale corpus is an effective way to enhance the performance of NLP systems. Volk used frequencies counts of query patterns to resolve PP attachment ambiguities via web search engine [5]. Lapata and Keller used web counts to resolve preposition attachments and compound noun interpretation [3][4]. Bansal used Google n-grams to generate full range of syntactic attachments [1]. Moreover, contextual statistics collected from web have also significantly improved the quality of automatic thesaurus extraction [6]. Yuan designed a series of rule-based scoring method for word categorization [7][8]. It was proven to be effective for validating parts of speech of Chinese words and categorizing phrases. Nevertheless, the principle of judging whether a word or a phrase follows a rule is completely based on decisions of linguistic experts. It requires significant amount of manual effort, which to a great extent weakens the scalability of patterns.

In section 2, we introduce a specialized search engine for lexicalized statistical patterns to conveniently acquire frequency information from any plain-text datasets. In this work, we experimentally collected Chinese plain-text sentences from the SogouT corpus. In section 3, we present some useful templates for Chinese analysis at different linguistic levels. For each level, the results of our case study are demonstrated respectively. Finally, we provide conclusion in Section 4.

2 Search Engine for Lexicalized Patterns

A naive way to obtain frequencies from web-scale corpus is to directly query from traditional web search engine. For example, if we query the phrase “告别朋友” using Google, it will return about 37,200 results. However, there are three main disadvantages of traditional search engine:

- There are duplicate pages and spam sites in the web environment. It will make the frequency counts unreliable.
- Web search engines ignore stop words and punctuation in general. But under some conditions we need these features to guarantee the quality of results.
- The users have no way to perform complex type of queries, such as near query, wildcards query or slop query.

For the example mentioned above, results from traditional search engine contain noises like “这是一场无声的告别, 朋友们再见!” (It’s a silent farewell. Goodbye friends!). The solution to this problem is to complement a period or a question mark behind the query, such as “告别朋友.” and “告别朋友?”. Moreover, we hope to permit the punctuation and query to be separated by no more than two words, like “告别朋友吧!” or “告别朋友了吗?”. To our knowledge, we are not aware of any web search engine that can deal with such kind of queries.

Hence, we manage to develop a flexible search engine to dig up lexicalized patterns more accurately. In fact, a similar product known as the Sketch Engine.³ has been published by Lexical Computing Ltd. It can deal with complex types of word queries, but it is not capable of handling Chinese documents without word segmentation.

In this paper, we propose to make use of Apache Solr⁴, which is an open source enterprise search platform. Solr provides document indexing APIs and support term proximity with slop factors. We implement a query builder to support formalized complex queries based on Solr. The unified regular expression of query is described as follow:

$$W(RW)*(RE)? \quad (1)$$

where W represents a candidate word list, R represents a range of wildcard gaps, and E represents a set of punctuation at the end of sentences. For example, one possible query to describe the pattern 告别朋友 can be written as:

$$\{\text{告别}\} <0-1> \{\text{朋友}\} <0-2> E \quad (2)$$

where “告别朋友吧!” and “告别朋友了吗?” both match this query.

In order to get the Chinese Web text corpus, we extract over 2.1 billion sentences from SogouT web page dataset, removing unnecessary html tags, hyperlinks, scripts and independent anchor texts. We take the number of matched sentences as the word count result for corresponding query pattern. To make sure the word count of our sketch search engine is reliable, we eliminate duplicates and short sentences (no longer than 2). We eventually index 729,008,561 unique sentences with a cost of 165.2 GB free disk space. The lexicalized search engine can handle all queries in the format described in (1), with an average of 10 seconds query response time. Three examples of word count statistics are demonstrated in Table 1.

3 Linguistical Analysis for Chinese Phrases

In this section, we analyze Chinese phrases at three linguistic levels: lexical, syntactic and semantic. The basic idea is to integrate handcrafted templates and then automatically acquire phrase counts via the search engine. For each level, we present the result of our case study on SogouT corpus respectively.

³ <http://www.sketchengine.co.uk/>

⁴ <http://lucene.apache.org/solr/>

3.1 Lexical Level

We focus on analyzing the cohesion of a compound noun phrase at lexical level. Phrases with low cohesion should not be included into vocabulary. For instance, 兔子尾巴 (rabbit tail) has low cohesion, while 圆桌会议 (round table) has high cohesion. Mutual information is commonly used to measure the cohesion of a phrase, but the boundary value between high and low cohesion is often fuzzy. For example, if we use SogouT to get frequency counts, the point-wise mutual information of 兔子 (rabbit) and 尾巴 (tail) is:

$$\begin{aligned} mi(\text{兔子}, \text{尾巴}) &= \log_2 \frac{N \cdot \text{Count}(\text{兔子尾巴})}{\text{Count}(\text{兔子}) \cdot \text{Count}(\text{尾巴})} \\ &= \log_2 \frac{729008561 \cdot 562}{171325 \cdot 193158} \\ &= 3.63 \end{aligned} \quad (3)$$

The result indicates that the mutual information of 兔子尾巴 is not relatively low. It is not enough to prove that this phrase has low cohesion.

In linguistic perspective, an important clue of high cohesion phrase is that the component words can hardly be separated by particles. Hence, we construct a template “NP₁+的+NP₂” to determine whether a compound noun phrase “NP₁ NP₂” has low cohesion. The statistic of lexicalized patterns are provided in the following:

$$\begin{array}{ll} \text{Count}(\text{兔子尾巴})= 562 & \text{Count}(\text{兔子的尾巴})= 272 \\ \text{Count}(\text{圆桌会议})= 6895 & \text{Count}(\text{圆桌的会议})= 8 \end{array}$$

The result shows that 兔子的尾巴 (the tail of rabbit) and 兔子尾巴 has a similar amount of occurrences, while 圆桌的会议 (the conference of round table) is significantly less frequent than 圆桌会议. In fact, 圆桌的会议 hardly appears in SogouT corpus. It indicates that the occurrence ratio of lexicalized patterns of NP₁+的+NP₂ is more effective than mutual information.

3.2 Syntactical Level

A typical problem at syntactic level is to determine the phrasal category of “VP NP” phrases. For example, 告别仪式 is a substantive phrase, while 告别朋友 is a verbal phrase. These two phrases share the same verb 告别 (say goodbye to). We can give many other examples, such as 修理公司 (repair company) and 修理自行车 (repair the bicycle), 学习小组 (study group) and 学习英语 (study English), 购买需求 (purchasing demand) and 购买基金 (purchase funds), etc.

Yuan (2010) distinguished five structural categories of compound phrases by designing handcrafted rules [8]. Inspired by these rules, we construct a set of templates for “VP NP” phrases via frequently-used auxiliaries and conjunctions in Chinese. For simplicity, we only take two-word compound phrase into account. We enumerate 14 templates in Table 2.

Table 2. Templates for determining the category of VP NP (do sth.) phrases

	Template	Translation	Example phrase
1	VP+了/着/过+NP	have done sth.	告别了朋友
2	不+VP+NP	do not do sth.	不告别朋友
3	VP+完/掉+NP	after doing sth.	告别完朋友
4	所+VP+的+NP	sth. done	所告别的朋友
5	把+NP+VP	to do sth.	把朋友告别
6	被+VP+的+NP	sth. that are done	被告别的朋友
7	为/拿+NP+VP	do for sth.	为朋友告别
8	VP+一+(量词)+NP	do a sth.	告别一个朋友
9	放着+NP+不+VP	sth. but not done	放着朋友不告别
10	VP+什么+NP	what sth. to do	告别什么朋友
11	NP+越+VP+越	the more one do sth.	朋友越告别越
12	连+NP+都/也+VP	don't even do sth.	连朋友都不告别
13	忙着+VP+NP	be busy doing sth.	忙着告别朋友
14	为什么+VP+NP	why one do sth.	为什么告别朋友

Given a “VP NP” phrase p , a lexicalized pattern $l(p, t)$ can be generated for each template t . If obtain the frequency counts via the search engine for lexicalized patterns, the phrase category of p is determined by the relative frequency $F(p)$, which is derived as follow:

$$F(p) = \frac{\sum_{t \in T} \text{Count}(l(p, t))}{\text{Count}(p)} \quad (4)$$

where $\text{Count}(l(p, t))$ represents the frequency count of each lexicalized pattern and $\text{Count}(p)$ represents the total count of phrase p . As we can see, the higher value of $F(p)$, the more likely the phrase p is to be verbal phrase. A simple validation algorithm is to set a lower threshold θ for verbal phrases, where θ may vary for different corpus.

To approximate the value of θ , we perform a case study on SogouT corpus. We collect 20 example phrases, half of which are verbal. The frequency results are demonstrated in Table 3. We find that the average phrase frequency of verbal phrases is only 60% higher than that of substantive phrases, while the average relative frequency $F(p)$ of verbal phrases is 52.3 times higher than that of substantive phrases. It indicates that the lexicalized patterns are effective to distinguish the two categories of phrases. Since the minimum $F(p)$ score among verbal phrases is 0.049 and the maximum $F(p)$ among substantive phrases is 0.006, the appropriate threshold θ for SogouT corpus can be set within the range of (0.006, 0.049).

Table 3. Frequency results of lexicalized patterns for verbal and substantive phrases. The translations of all these phrases are given in Appendix.

Category	Phrase	Phrase frequency	Pattern frequency	$F(p)$
Verbal	告别朋友	116	99	0.853
	购买基金	9508	597	0.063
	修理自行车	752	40	0.053
	学习英语	27763	1361	0.049
	治疗病人	1453	334	0.230
	救济灾民	897	13	0.014
	判罚点球	5870	1373	0.234
	维护秩序	3972	173	0.044
	攻击敌人	9337	357	0.038
	更新系统	1504	147	0.098
	Average	5617.2	449.4	0.080
Substantive	告别仪式	8460	1	0.0001
	购买需求	4750	5	0.0001
	修理公司	1167	7	0.005
	学习小组	5079	23	0.005
	治疗手段	11273	8	0.0001
	救济中心	148	0	0.000
	判罚尺度	1222	1	0.0001
	维护工具	1107	0	0.000
	攻击计划	704	2	0.003
	更新日期	832	5	0.006
	Average	3474.2	5.2	0.0015

3.3 Semantic Level

At the semantic level, we focus on discovering the patient object, which is a nominal phrase that acts as the recipient of the action stated by a verbal phrase. It is a significant semantic relation between nominal and verbal phrases. For example, 晚饭 (dinner) is the patient of verb 吃 (eat) in the phrase 吃晚饭 (having dinner), while 食堂 (canteen) is not the patient of 吃 (eat) in the phrase 吃食堂 (eating at the canteen).

Given a predicate-object phrase “VP NP”, the task of identifying the patient object usually rely on the judgement of linguists. In this paper, we propose to utilize a set of templates for this task, such as “把+NP+VP+了”, “所+VP+的+NP” and “被+VP+的+NP”. If the nominal phrase occur to be the patient object, the lexicalized pattern for this template tends to occur frequently in web-scale corpus. For example, 把晚饭吃了 (to have dinner) appears 63 times in SogouT, while 把食堂吃了 (to eat the canteen) has never appeared. Here, we present the frequencies of 8 examples in SogouT to verify the reliability of our template.

Patient objects:

Count(吃晚饭)= 16608	Count(把晚饭吃了)= 63
Count(炒菜) = 31473	Count(把菜炒了) = 18
Count(洗衣服)= 50989	Count(把衣服洗了)= 353
Count(割阑尾)= 157	Count(把阑尾割了)= 4

Non-patient objects:

Count(吃食堂)= 1798	Count(把食堂吃了)= 0
Count(去北京)= 6895	Count(把北京去了)= 0
Count(想主意)= 919	Count(把主意想了)= 0
Count(筹经费)= 4377	Count(把经费筹了)= 8

where the translations of these phrases are given in Appendix.

As we can see, for patient object, the frequency of lexicalized pattern is proportional to the frequency of the VP+NP phrase. For non-patient object, the frequency of lexicalized pattern is almost always zero. It indicates that our template is effective for discovering patient objects among VP+NP phrases.

4 Conclusion

We exploit lexicalized statistical patterns collected from the web corpus at three linguistic levels of Chinese analysis. Results of our case study indicates that these patterns are effective on analyzing the cohesion of phrases, determining the phrasal category and discovering patient objects.

Our automatic categorization of phrasal category contributes to reduce the workloads of linguistics [7][8]. The search engine for lexicalized patterns can also be used for verifying the effectiveness of batched and hand-crafted linguistic rules. In the future we aim at integrating the lexicalized statistical patterns as feature templates to enhance the precision of Chinese parser.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant No. 61133012) and the National High Tech. Development Program of China (863 Program) (Grant No. 2012AA011102).

References

1. Bansal, M., Klein, D.: Web-scale features for full-scale parsing. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (2011)
2. Curran, J.R., Moens, M.: Scaling context space. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA (2002)
3. Keller, F., Lapata, M., Ourioupina, O.: Using the web to overcome data sparseness. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia (2002)

4. Lapata, M., Keller, F.: The Web as a baseline: Evaluating the performance of unsupervised Web-based models for a range of NLP tasks. In: Proceedings of HLT-NAACL (2004)
5. Volk, M.: Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In: Proceedings of the Corpus Linguistics 2001 Conference, Lancaster, UK, pp. 601–606 (2001)
6. Yates, A., Schoenmackers, S., Etzioni, O.: Detecting parser errors using web-based semantic filters. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2006)
7. Yuan, Y.: A Cognitive Investigation and Fuzzy Classification of Word-class in Mandarin Chinese. Shanghai Educational Publishing House (2009)
8. Yuan, Y.: 汉语词类划分手册, Beijing Language and Culture University Press (2010)
9. Zhang, Y., Clark, S.: Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics* 37(1), 105–151 (2011)

Appendix

Here we present the English glosses of some Chinese phrases in this paper to make sure one has a clearer understanding of their meanings.

告别朋友	to farewell a friend	告别仪式	the farewell ceremony
购买基金	to purchase funds	购买需求	the purchasing need
修理自行车	to repair the bicycle	修理公司	the repair company
学习英语	to study English	学习小组	the study group
治疗病人	to cure patients	治疗手段	the treatment
救济灾民	to relieve the victims	救济中心	the relief center
判罚点球	to give a penalty	判罚尺度	the principle of decision
维护秩序	to maintain order	维护工具	the maintenance tool
攻击敌人	to attack the enemy	攻击计划	the attacking plan
更新系统	to update the system	更新日期	the update date
吃晚饭	have dinner	炒菜	cook dishes
洗衣服	wash clothes	割阑尾	cut the appendix
吃食堂	eat at the canteen	去北京	go to Beijing
想主意	think of ideas	筹经费	raise money

Development of Traditional Mongolian Dependency Treebank

Xiangdong Su, Guanglai Gao, and Xueliang Yan

College of Computer Science
Inner Mongolia University
Huhhot China 010021
csggl@imu.edu.cn

Abstract. This paper describes the development of Traditional Mongolian dependency treebank (TMDT) which aims to facilitate the dependency analysis on Traditional Mongolian. The annotation scheme of the dependency treebank is established according to Traditional Mongolian grammar and its usability in syntactic analysis. In the treebank, morphological and analytical information are annotated. At morphological level, a semi-automation strategy is adopted. Part-Of-Speech (POS) and stem of each word in the sentence are tagged and extracted respectively with automation tools, and then manually corrected. At analytical level, the dependencies in the sentence are only annotated manually according to constituent structure and the annotation scheme. This treebank formulates the foundation of dependency parsing on Traditional Mongolian and can be extended to a multi-dependency Treebank.

Keywords: Traditional Mongolian, Dependency Treebank, Morphological Information, Analytical Dependency.

1 Introduction

Syntactic parsing is always the active research area in natural language processing. In this field, much progress has been made in its theories and several applicable systems have been developed. As a subfield of syntactic parsing, dependence parsing recently gains more and more attention among researchers since it provides useful information in many document-analysis related applications. In dependency parsing, dependency treebank is required to train the parser and evaluate its performance. So far, treebanks have been constructed for many languages, including English, French, German, Spanish, Turkish, Russian, and so on. To facilitate the dependency analysis in Traditional Mongolian, we develop a Traditional Mongolian dependency treebank (TMDT) on the basis of in-depth study of Traditional Mongolian grammars and its usability in syntactic analysis.

As one Asian language, Traditional Mongolian is derived from Uyghur and used in Inner Mongolia and neighboring regions. It has over 5 million speakers. Traditional Mongolian possesses agglutinative word structure with complex inflection and derivation. Its word is consisted of letters which are connected along a straight line, called

spine. Each letter has as many as three different shapes depending on whether the letter appears in an initial, medial, or final position. In some cases, additional graphic variants are selected for visual harmony with the subsequent letter. From the syntactic viewpoint, Traditional Mongolian has SOV constituent order and the predicate is not necessary to be a verb or copula. There are totally eight kinds of case in Traditional Mongolian. Case markings on nominal constituents usually indicate their syntactic role. Some cases act as preposition or conjunction. We take the above mentioned characteristics into consideration in building the dependency treebank.

This paper mainly describes the annotation scheme of the treebank, and simply reports the annotation procedure as well as the final production. TMDT takes a two-level annotation structure including morphological level and analytical level. Morphological level annotates the POS and stem of each word in the sentence, considering their importance for Traditional Mongolian word. Analytical level annotates the binary dependency relationship holding between a syntactically subordinate word, called child, and the other word on which it depends, called parent. Morphological information contributes to the analytical annotation. The annotation process is divided into two phases according to the annotated information: morphological annotation phase and analytical annotation phase. In morphological phase, a semi-automation strategy is employed to speed up the annotation process. In the analytical phase, the dependency relationships are only annotated manually without any automaton's assistance. The whole Treebank is reviewed carefully to ensure its correctness.

The rest of this paper is structured in the following way: Section 2 mentions some related work about dependency treebank development. Section 3 describes the annotation scheme including the annotation principles, structure and tag set. Section 4 reports the annotation workflow, treebank format and final production. An example is provided to intuitively present the resulting treebank. Finally, section 5 concludes this paper and points out the future direction.

2 Related Work

Much past work is related with treebank building. M.P. Marcus et al. in [1] present the influential treebank, penn treebank, which leads the way in building annotated treebank and serves as an excellent model for treebank building of Traditional Mongolian. B. Rajesh et al. in [2] depict the creation of Hindi/Urdu multi-representational and multi-layered treebank. A. Böhmová et al. in [3] describe a three level annotation scheme in building the Prague dependency treebank, including Morphological level, analytical level and tectogrammatical level respectively. They manually annotate the dependency treebank, and automatically generate the phrase structure tree bank from the dependency treebank. C.-R. Huang et al. in [4] specify the design criteria and annotation guidelines of Sinica treebank. The three design criteria are: Maximal Resource Sharing, Minimal Structural Complexity, and Optimal Semantic Information. P. Pajas and J. Štěpánek in [5] propose an annotation framework that was designed to be extensible and independent of any particular annotation schema. M.-C.d. Marneffe and C.D. Manning in [6] examine the Stanford typed dependencies representation, which was designed to provide a straightforward description of grammatical relationships.

In dependency treebank annotation, the core task is discerning the dependency relationships which vary with languages and dependency grammars. I.A. Melčuk in [7] discusses morphological, syntactic and semantic dependencies in Meaning-Text theory. R. Hudson in [8] details the English dependency theory. J. Nivre in [9] reviews the dependency detection criteria.

Manually and semi-automation annotations are the mainstream of annotation strategy. T. Brants et al. in [10] explore (1) the automation of Treebank annotation, (2) the comparison of conflicting annotations and (3) the inconsistencies detection in automatic annotation. L.v.d. Beek et al. in [11] use Alpino parser and parse selection tool to facilitate the annotation process of Alpino dependency treebank.

The following works are also related to our work. J. Lafferty et al. in [12] view POS tagging as a sequence labeling problem and obtain a better performance with CRFs tagger. M.-Y. Ma in [13] brings forward a mixed model for Traditional Mongolian Stemming. W.-B. Jiang et al. in [14] employ Lexical Analyzer based on directed graph to segment the stem and affix of Mongolian word. More information about Traditional Mongolian Lexical and syntactic grammar can be found in the book [15].

3 Annotation Scheme

So far, most of the available dependency treebanks take into account both morphological and analytical information. Yet Prague dependency treebank encodes the dependencies at semantic level. In TMDT, we just annotate the sentences at two levels: morphological level and analytical level. Firstly, the morphological captures the basic attributes of the syntactic units (words) and helps to the annotation at analytical level. The analytical level specifies the dependency information of the sentences. Secondly, semantic annotation is very complex process and relates to the deep structure of the sentence. We take no account of the semantic dependency in TMDT since there are still some disagreements among researches about the semantic relationship in Traditional Mongolian. Furthermore, the criterion of minimal structural complexity is adopted to ensure that the assigned structural information can be used without any assumption about the user's background. We will deal with the annotation levels in turn, starting with morphological level in section 3.1 and continuing with the analytical level in section 3.2.

3.1 Morphological Level

Morphological information expresses the attributes of the syntactic units and plays an important role in syntactic analysis. In TMDT, The morphological annotation principles are as follows.

1. Annotation unit

As mentioned above, many words in Traditional Mongolian are produced through inflection and derivation. The change in the form of a word (typically the ending) usually expresses a grammatical function or attribute such as tense, mood, number, case, and gender. We treat the derivative as annotation unit and take no consideration of the

inflection and derivation phenomenon, except the case inflection phenomenon. For instance, adding a tense suffix "ᠠᠭᠢ", "ᠠᠭᠢᠨ" (whose meaning is "go" in English) becomes "ᠠᠭᠢᠨᠠᠭᠢ" (whose meaning is "have gone" in English). "ᠠᠭᠢᠨᠠᠭᠢ" is treated as a basic annotated unit. Although our strategy increases the amount of lemmas in dictionary, it simplifies the annotation process and treebank representation.

2. Case inflection

Case inflection is the phenomenon that adding a case to a noun, adjective, or pronoun that express the semantic relation of the word to other words in the sentence. Some cases are separated with the previous word they attach to by a common blank (Unicode 0X0020). The other cases are separated with the previous word they attach to by Mongolian blank (Unicode 0X202F). In Traditional Mongolian grammar, the attached word and the case in the latter situation are considered as a single word. We take a different perspective and treat them as two annotation units considering the case function in sentences. This is more suitable to practical application.

3. Annotating Part-Of-Speech (POS) tag

POS are known as word classes or lexical categories and greatly related to the constituent role of syntactic unit. So POS is annotated at morphological level. POS tagging is the process of classifying words into their POS. The collection of POS tags used in TMDT is listed in Table 1.

4. Annotating word's stem

In computational linguistics, each word in Traditional Mongolian is made up of a solo stem or a stem with one or more suffixes. The stem of a word is the part which is common to all its inflected variants. This means that the stem is an essential attribute of Traditional Mongolian word and represents the original meaning of the word. For example, "ᠰᠢᠨᠠᠭᠢᠨᠠᠭᠢ" is a new word produced by adding "ᠠᠭᠢᠨᠠᠭᠢ" to "ᠰᠢᠨᠠ". Here, "ᠰᠢᠨᠠ" is the stem, and "ᠠᠭᠢᠨᠠᠭᠢ" is the suffix. Therefore, we integrate the word's stem into the target dependency treebank.

Table 1. POS Tags of Traditional Mongolian in TMDT

Category	Description	Category	Description
NN	noun	NUM	number
QUAN	quantity	PRON	pronoun
ADJ	adjective	ADV	adverb
VB	verb	POSS	Possecive pronoun
CONJ	conjunction	MOD	modal verb
MOOD	mood word	WH	interrogative word
LEXAUX	combining form	AUXI	auxiliary word
PUNC	punctuations	FORW	foreign words
NOMCA	nominative case	ACCA	accusative case
REFCA	reflexive case	INSCA	instrumental case
GENCA	genitive case	DLCA	dative-locative case
ABLCA	ablative case	COMCA	comitative case
TEPO	temporal and positional words	REFVB	special words link thinking and speech

Table 2. Categories of Analytical Dependency in TMDT

Category	Description
clau	dependency between main clause and subordinate clause
indcla	dependency between main clause and independent constituent
nsubj	dependency between predict and subject
nobj	dependency between predict and object
modi	dependency between the modifier and the object been modified
aux	dependency between the auxiliary and the object it act on
advmod	dependency between the adverbial modifier and predict
nomca	dependency between the true subject and nominative case
acca	dependency between the true object and accusative case
genca	dependency between the modifier and genitive case
ablca	dependency between the object of adverbial phrase and ablative case
dlca	dependency between the object of adverbial phrase and dative-locative case
insca	dependency between the object of the adverbial phrase and instrumental case
comca	dependency between the object of adverbial phrase and comitative case
refl	dependency between the referenced content and REFVB
conj	dependency between the conjunction and the first object it linked
coord	dependency between the coordination or appositional components
lex	lexical relation between two words
punct	dependency between the punctuation and the part it act on
dep	dependency that is unable to determine a more precise relation
root	root

In TMDT, 21 kinds of analytical dependency are defined according to the annotation principles and the syntactic characters of Traditional Mongolian. Table 2 lists all the dependencies together with their descriptions.

4 Treebank Building

This section describes the annotation workflow, treebank format and final production in TMDT building.

4.1 Annotation Workflow

Fig. 1 shows the annotation workflow in treebank development. To speed up the annotation process, we use a CRF tagger proposed in [12] to label POS tags and implement the algorithm (StemExtractor) proposed in [13] to extract the stem for each word. However, purely automatic annotation without supervision is not reliable. Therefore, we manually correct POS tag and stem resulting from the automation tools

subsequently. The dependency relationships at the analytical level are just annotated manually without any automaton's assistant. In this process, two annotators independently choose analytical tag from the dependency list defined in the annotation scheme and use an annotator-arbiter strategy to resolve the conflicts between them. This ensures the correctness of the target dependency treebank.

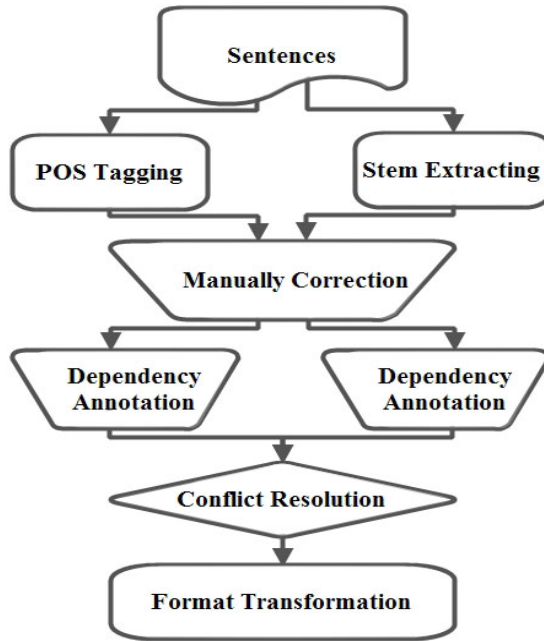


Fig. 1. Workflow of the Annotation Procedure

4.2 Output Format and Final Production

Two types of treebank formats were employed in TMDT. The first format, which is similar to CoNLL-08, is demonstrated in Table 3. The first column represents the orders of syntactic units in sentences, and the second column represents the syntactic units. The remnants are stem, POS tag, dependency category and head respectively. The fifth column represents the dependency between the heads whose sequence is the corresponding number and the dependent which is the word in the corresponding row. TMDT can be extended to multi-dependency treebank. The output format is just adding a “rel” column and a “head” column in the first format. Table 3 is the first format of the sentence "Агаарыг / шалгахыг / нь / өндөр / гүйцэтгэл / нь / бусдын / хувь / хүчин / гүйцэтгэл / нь /". The second format is XML because of its popularity, its ease of understanding and its wide use in description of linguistic information. It is automatically transformed from the first format.

The annotated corpus comes from Inner Mongolian daily. The content involves all aspects of the social life. These sentences conform to the grammar, and are few of mistakes. This is convenient for treebank annotation and exploiting the linguistic phenomenon in Traditional Mongolian. The annotation corpus is in Mongolian Universal Coding.

TMDT is composed of 400 sentences (13028 annotated words) from Inner Mongolian daily. Among these sentences, the shortest one contains 9 words and the longest one contains 54 words. The average length is 32.57 words. Simple sentences account for 27%, and clauses account for 73%. There are 4460 distinct words and 1548 distinct stems. The non-projective trees account for 2.25% in the whole treebank. Fig. 2 shows the annotation result of "ᠠᠯᠠᠳᠤ ᠠᠯᠤ ᠪᠢᠶ᠋ᠠᠨᠲᠤᠨᠠᠵᠢ ᠰᠤ ᠪᠢᠶ᠋ᠠᠨᠲᠤ ᠶ᠋ᠢᠨᠠᠨᠠᠳᠤ ᠪᠢ ᠪᠠᠶ᠋ᠢᠨᠠᠳᠤ ᠠᠬᠤ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ" at morphological level and analytical level in TMDT.

Table 3. The first format of "ᠠᠯᠠᠳᠤ ᠠᠯᠤ ᠪᠢᠶ᠋ᠠᠨᠲᠤᠨᠠᠵᠢ ᠰᠤ ᠪᠢᠶ᠋ᠠᠨᠲᠤ ᠶ᠋ᠢᠨᠠᠨᠠᠳᠤ ᠪᠢ ᠪᠠᠶ᠋ᠢᠨᠠᠳᠤ ᠠᠬᠤ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ"

seq.	word	stem	POS	rel	head
1	ᠠᠯᠠᠳᠤ	ᠠᠯᠠᠳᠤ	NN	poss	2
2	ᠠᠯᠤ	ᠠᠯᠤ	POSSCA	modi	3
3	ᠪᠢᠶ᠋ᠠᠨᠲᠤᠨᠠᠵᠢ	ᠪᠢᠶ᠋ᠠᠨ	NN	subca	4
4	ᠰᠤ	ᠰᠤ	SUBCA	nsubj	11
5	ᠪᠢᠶ᠋ᠠᠨᠲᠤ	ᠪᠢᠶ᠋ᠠᠨ	NN	poss	7
6	ᠶ᠋ᠢᠨᠠᠨᠠᠳᠤ	ᠶ᠋ᠢᠨᠠᠨᠠᠳᠤ	NN	lex	5
7	ᠪᠢ	ᠪᠢ	POSSCA	modi	8
8	ᠪᠠᠶ᠋ᠢᠨᠠᠳᠤ	ᠪᠠᠶ᠋ᠢᠨᠠᠳᠤ	NN	objca	10
9	ᠠᠬᠤ	ᠠᠬᠤ	NN	lex	8
10	ᠠᠨᠠᠵᠢ	ᠠᠨᠠᠵᠢ	OBJCA	nobj	11
11	ᠠᠨᠠᠵᠢ	ᠠᠨᠠᠵᠢ	VB	root	0
12	ᠠᠨᠠᠵᠢ	ᠠᠨᠠᠵᠢ	AUXI	aux	11
13	ᠠᠨᠠᠵᠢ	ᠠᠨᠠᠵᠢ	PUNC	punct	11

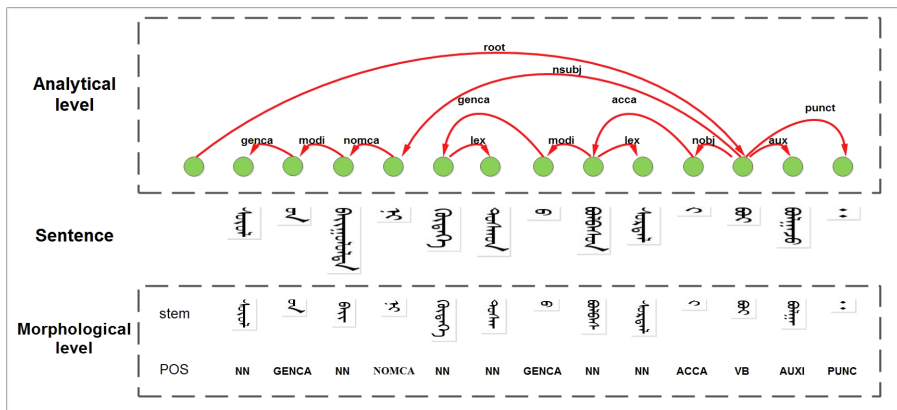


Fig. 2. Annotation Scheme of "ᠠᠯᠠᠳᠤ ᠠᠯᠤ ᠪᠢᠶ᠋ᠠᠨᠲᠤᠨᠠᠵᠢ ᠰᠤ ᠪᠢᠶ᠋ᠠᠨᠲᠤ ᠶ᠋ᠢᠨᠠᠨᠠᠳᠤ ᠪᠢ ᠪᠠᠶ᠋ᠢᠨᠠᠳᠤ ᠠᠬᠤ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ ᠠᠨᠠᠵᠢ" in TMDT

5 Conclusion and Future Work

A dependency treebank is very important in syntactic analysis. However, there is no suitable dependency treebank for Traditional Mongolian. For this reason we start to develop Traditional Mongolian dependency treebank by annotating the corpus coming from Inner Mongolian daily with dependency structures. This paper describes the annotation scheme, annotation procedure and the final production. This treebank is annotated at morphological level and analytical level, which labels the POS tags, word's stem and the syntactic dependency relationships. Our work yields promising results, indicating the annotation scheme of TMDT treebank is essential to the success of building a multi-layered treebank. This treebank can be extended to a multi-dependency treebank, in which many types of dependency relationship co-exist between two syntactic units in sentences. This work formulates the foundation of dependency parsing on Traditional Mongolian. This is not an end, but rather a roadmap to the syntactic analysis on Traditional Mongolian, with some progress along the way, since the theories of linguistics is still in development.

In the future, we will continue to expand the treebank's scale and attempt to convert it into a phrase structure treebank using statistical methods.

Acknowledgments. This work is supported by National Natural Science Foundation of China (Grant No. 61263037) and Major Program of Natural Science Foundation of Inner Mongolia of China (Grant No. 2011ZD11).

References

1. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19, 313–330 (1994)
2. Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D.M., Xia, F.: A Multi-representational and Multi-layered Treebank for Hindi/Urdu. In: *Proceedings of the Third Linguistic Annotation Workshop*, pp. 186–189. Association for Computational Linguistics, Suntec (2009)
3. Böhmová, A., Hajič, J., Hajičová, E., Hladká, B.: The Prague Dependency Treebank: A Three-Level Annotation Scenario. In: Abeillé, A. (ed.) *Treebanks: Building and Using Syntactically Annotated Corpora*, pp. 103–127. Kluwer Academic Publishers (2001)
4. Huang, C.-R., Chen, F.-Y., Chen, K.-J., Gao, Z.-M., Chen, K.-Y.: Sinica Treebank: Design Criteria, Annotation Guidelines, and On-line Interface. In: *Second Chinese Language Processing Workshop*, pp. 29–37. Association for Computational Linguistics, Hong Kong (2000)
5. Pajas, P., Štěpánek, J.: Recent Advances in a Feature-Rich Framework for Treebank Annotation. In: *Proceedings of the 22nd International Conference on Computational Linguistics*, vol. 1, pp. 673–680. Association for Computational Linguistics, Manchester (2008)
6. 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*, pp. 1–8. Association for Computational Linguistics, Manchester (2008)
7. Mel'čuk, I.A.: *Dependency Syntax: Theory and Practice*. State University of New York Press, New York (1988)

8. Hudson, R.: *An Introduction to Word Grammar*. Cambridge University Press, Cambridge (2010)
9. Nivre, J.: *Dependency Grammar and Dependency Parsing*. Technical Report, School of Mathematics and Systems Engineering, Växjö University (2005)
10. Brants, T., Skut, W.: *Automation of Treebank Annotation*. In: *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, pp. 49–57. Association for Computational Linguistics, Sydney (1998)
11. van der Beek, L., Bouma, G., Malouf, R., van Noord, G.: *The Alpino Dependency Treebank*. *Computational Linguistics in the Netherlands, CLIN* (2002)
12. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. In: *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289. Morgan Kaufmann Publishers Inc. (2001)
13. Ma, M.-Y.: *Researching of Mongolian Word Segmentation System Based on Dictionary, Rules and Language Model*. Computer Science, Inner Mongolian University, master (2011) (in Chinese)
14. Jiang, W.-B., Wu, J.-X., Wurliga, Nashunwritu, Liu, Q.: *Discriminative Stem-Affix Segmentation for Directed-Graph-Based Mongolian Lexical Analyzer*. *Journal of Chinese Information Processing* 25, 30–34 (2011)
15. Qinggeertai: *Traditional Mongolian Grammar*. Inner Mongolian Press, Huhhot (1992) (in Chinese)
16. König, E., Lezius, W.: *The TIGER Language: A Description Language for Syntax Graphs, Formal Definition* (2003)

Chinese Sentence Compression: Corpus and Evaluation

Chunliang Zhang, Minghan Hu, Tong Xiao, Xue Jiang, Lixin Shi, and Jingbo Zhu

Natural Language Lab, Northeastern University, Shenyang, China, 110819
{zhangcl, huminghan, xiaotong, zhujiangbo}@mail.neu.edu.cn,
{neujiangxue, shilixin}@163.com

Abstract. In this paper we present a first-ever manually-built Chinese sentence compression corpus. Based on this corpus, we develop a Chinese sentence compression system and study various measures for evaluation of Chinese sentence compression. We find that 1) using multi-references is very helpful for automatic evaluation in Chinese sentence compression; and 2) besides relational F1, some machine translation evaluation measures are correlated well with human judgments and thus are very promising for future use in this task.

Keywords: sentence compression, Chinese corpus, system evaluation.

1 Introduction

Recent years have seen increasing interests in automatic sentence compression among the natural language processing researchers for a wide range of practical applications, such as text summarization, machine translation, and question answering. In general, the task of sentence compression can be described as creating a shorter form of a sentence while retaining the most important information and remaining grammaticality [8]. To date, many statistical models have been developed in sentence compression, showing continuous improvements on several tasks ([3-4], [7], [10-11], [18], [21-22]).

Naturally, as with any NLP technique, developing sentence compression systems relies on manually annotated corpora for training model parameters (in a supervised manner), system tuning, and evaluation of final results. However, the scarcity of such data restricts most work in English compression tasks (e.g., the Ziff-Davis corpus) and it is rare to see efforts in other languages.

In this paper we study the sentence compression problem for Chinese, one of the most popular languages other than English. We regard sentence compression as a task of identifying the selection word sequence of a sentence. In this way a compressed sentence is in principle a backbone of the original sentence and can be generated by removing all “unimportant” words. The contributions of this work are two-fold:

- We manually develop a Chinese sentence compression corpus consisting of 3,308 sentences from the Penn Chinese Treebank. For each sentence, there is at least one annotation. In addition, we provide three annotations for a sub-set of 563 sentences, which can be used as benchmark for evaluation of Chinese sentence compression systems. To the best of our knowledge, this is the first-ever manually annotated corpus for Chinese sentence compression.

- We study various evaluation metrics on the developed system for Chinese sentence compression. We find that 1) using multiple references is more helpful for automatic evaluation of the system performance than using the single-reference, as the strategy adopted in previous studies; 2) four evaluation metrics, grammatical relations F1, mWER, mPER and GTM, have good co-relations with human judgments when used to measure the performance of the compression systems in terms of grammaticality and importance, and, therefore, are more desirable for automatic evaluation of Chinese sentence compression.

More importantly, our annotated corpus (as well as the annotation guideline, the automatic system output and human judgments) is accessible to public¹ and can be used in further study and system development for this task. We believe that the developed corpus would motivate more studies on identifying the skeleton/main structure of Chinese sentences and would thus benefit many downstream NLP applications, such as machine translation and text summarization.

2 Related Work

Most previous work addresses the sentence compression task on English corpora. The most famous of these is the Ziff-Davis corpus [9], a collection of 1,067 sentences created automatically by matching sentences in a news article with sentences contained in its abstract. Yamangil and Nelken [20] collected a large-scale corpus of over 380,000 sentence pairs by mining the Wikipedia revision history of the articles and picking out those sentences with the record of word addition or deletion. But their work was based on an assumption that all the edits retain the core meaning of the sentence. There are two corpora manually created for English sentence compression [2], one is a 1,433-sentence dataset built from the British National Corpus and the American News Text Corpus, and the other is a 1,370-sentence dataset from the HUB-4 1996 English Broadcast News Corpus. However, to our knowledge, there is no such data in Chinese for sentence compression research.

3 Corpus Development

3.1 Data Selection

The original data in this work comes from the source-language side of the Penn Parallel Chinese-English Treebank (LDC2003E07). We choose this data set for annotation because all the sentences in the Penn Chinese Treebank (CTB) are of very good quality [19]. As these CTB sentences have been manually annotated with word segmentation, POS tags and syntactic structures, we believe they will be useful in studying the sentence compression problem on different conditions, e.g., comparison of the results obtained on gold-standard and automatic word segmentations/syntactic trees. Besides, our dataset in Chinese parallels with its English counterpart, and thus can be used in future studies of bilingual sentence compression or applying compression results in machine translation.

¹ <http://202.118.18.77:8080/ChineseSentenceCompression/>

For convenience of annotation, we divide the selected dataset into 10 parts. Parts 1-8 consist of articles 001-270 and are with one annotation. Part 9 and Part 10 consist of articles 271-300 and articles 301-325 respectively, and both are with three annotations.

Table 1. The dataset used for annotation

Dataset	# Sentences	# Words	# Annotators
All (#1-3308)	3308	74312	1-3
Parts 1-8 (#1-2745)	2745	62868	1
Part 9 (#2746-3018)	273	5131	3
Part10 (#3019-3308)	290	6313	3

3.2 Annotation Guideline

This study focuses on Chinese sentence compression mainly for identifying the main structure of the Chinese sentence. We view sentence compression as a task of keeping the most important grammatically-motivated items of a sentence and removing all unimportant items. So in this work the result of sentence compression is in essential a grammatically-motivated skeleton of the input sentence.

In creating annotations for sentence compression, annotators are provided with the sentences only with word segmentation² and are required to compress the sentences by deleting the unimportant words while remaining sentence grammaticality.

Similar to the English counterpart, a Chinese sentence is composed of several constituents: the subject, the predicate, the object, the attributive, the adverbial, and the complement. The subject, the predicate, and the object are primary constituents, and the attributive, the adverbial, and the complement are secondary ones.

Original: <晌午> 的 <太阳> <火辣辣> 地 <烤> 着 <田野> 。 <noon>DE1<sun> <fiery> DE2<scorch>ZHE <field> attr. sub. adv. pred. obj. (The sun is scorching the field like fire at noon.)
Compressed: <太阳> <烤> 着 <田野> 。 <sun> <scorch> ZHE <field> sub. pred. obj. (The sun is scorching the field.)

Fig. 1. A demo of Chinese sentence compression with the sentence constituent analysis

² Word segmentation is a necessary step for most natural language processing tasks on Chinese for there is no delimiter between Chinese words. In this task, the information other than word segmentation is not available for annotators.

Basically, to achieve sentence compression, the first thing is to identify different constituents in a sentence, and then to remove the secondary constituents and retain the primary ones, because we believe the primary constituents constitute the structural backbone and carry the most valuable information in a sentence, and the secondary constituents just act as modifiers of one primary constituent and carry unimportant information. As shown in Fig.13, the annotators should first decompose the sentence into different constituents: the subject(sub.), the predicative(pred.), the object(obj.), the attributive(ATTR.), and the adverbial(adv.). Note that this Chinese sentence includes three auxiliary words: two structure-auxiliary words⁴ ‘的(DE1)’ and ‘地(DE2)’, which mark the preceding constituents as the attributive and the adverbial respectively, and one aspect-auxiliary word ‘着(ZHE)’⁵, which is attached to the preceding verb ‘烤(scorch)’ and acts as the indicator of the durative aspect for the verb. After the sentence constituents are identified, the sentence compression is done by deleting the attributive ‘晌午(noon)’, the adverbial ‘火辣辣(fiery)’ and their attached structural auxiliary words, ‘的(DE1)’ and ‘地(DE2)’.

In practice, the word deletion is done at two levels: the word level and the phrase level. To save space, we list only a few critical annotation rules here⁶.

At the word level, all the adjectives will be deleted if they modify a noun/noun phrase, as the phrase⁷ ‘<高昂>的 <成本>(high)DE1<cost>, high cost)’ is compressed as a noun ‘<成本>(cost)’ by deleting the adjective ‘<高昂>(high)’ and the structural auxiliary ‘的(DE1)’. Besides, the degree adverbs, such as ‘很(very)’ and ‘有点儿(a little)’, will be deleted if they modify an adjective, as the phrase ‘很美(very beautiful)’ is compressed as an adjective ‘美(beautiful)’.

<p>Original: <据 报道>, 朝鲜 代表团 已经 抵达 北京。 According to report, DPRK delegation already arrive Beijing. (It is reported that the DPRK delegation has arrived at Beijing.)</p> <p>Compressed: 代表团 已经 抵达 北京。 delegation already arrive Beijing. (The delegation has arrived at Beijing.)</p>
--

Fig. 2. A demo of the parenthesis deletion for Chinese sentence compression

³ The double vertical lines ‘||’ in Fig. 1 shows the boundary between the subject and the predicate, the two primary constituents in a sentence.

⁴ The Chinese auxiliary words, ‘的’, ‘地’, and ‘得’, are usually denoted as ‘DE1’, ‘DE2’ and ‘DE3’ respectively in analysis of the syntactic structure.

⁵ The Chinese auxiliary words, ‘着(ZHE)’, ‘了(LE)’, ‘过(GUO)’, are attached to a verb to mark its aspect and tense.

⁶ For detailed description of the annotation guideline, please refer to <http://202.118.18.77:8080/ChineseSentenceCompression/>.

⁷ The ‘phrase’ used here, instead of the ‘sentence’ is for space-saving. It is by no means to compress a phrase in this work.

Table 2. A demo of some fundamental annotation rules for Chinese sentence compression

Comp. target	Example
adjectives	original: <美丽的> <蝴蝶> <飞走> <了>。 (The <i>beautiful</i> butterfly flew away.)
	compressed: <蝴蝶> <飞走> <了>。 (The butterfly flew away.)
degree adverbs	original: <这里的> <景色> <真美>。 (The scenery here is <i>really</i> beautiful.)
	compressed: <景色> <美>。 (The scenery is beautiful.)
noun phrases	original: <中国 国家 主席 习近平> <将> <于近日> <出访> <俄罗斯>。 (The <i>Chinese president Xi Jinping</i> will visit Russia in a few days.)
	compressed: <习近平> <将> <出访> <俄罗斯>。 (Xi Jinping will visit Russia.)
prep. phrases	original: <老师> <希望> <我们> <为了美好的明天> 而 <学习>。 (The teacher hopes we will study for a beautiful tomorrow.)
	compressed: <老师> <希望> <我们> <学习>。 (The teacher hopes we will study.)
parentheses	original: <在 中国 的 大 城市>, <尤其是北京 和 上海>, <交通 堵塞> <非 常 严重>。 (In large cities in china, <i>especially Beijing and Shanghai</i> , traffic jam is very serious.)
	compressed: <交通 堵塞> <严重>。 (The traffic jam is serious.)

At the phrase level, the compression rules are mainly concerned about the noun phrase and the prepositional phrase. For the noun phrase comprised of a succession of nouns, some nouns will be deleted if they modify the other nouns, as the noun phrase ‘美国 有线新闻网 记者(CNN correspondent)’ is compressed as the noun ‘记者(correspondent)’, for human annotators can easily distinguish the two units of the phrase, the unit of a proper noun, ‘美国 有线新闻网(CNN)’, modifying the unit of a noun ‘记者(correspondent)’. Another kind of noun phrase is that it contains two coreferents, like the phrase ‘中国 国家主席 习近平(Chinese president Xi Jinping)’, where ‘中国 国家主席(Chinese president)’, the job title, and ‘习近平(Xi Jinping)’, the person’s name, corefer to each other. In such a case, the compression is done by deleting one of them (usually retaining the proper noun). For the prepositional phrase, it will be deleted when it functions as the adverbial in the sentence, as the phrase ‘为了美好的明天而学习(study for a beautiful tomorrow)’ is compressed as ‘学习(study)’ by deleting the prepositional phrase ‘为了美好的明天(for a beautiful tomorrow)’.

Besides the above rules, the parenthesis⁸ in a sentence will be deleted during the compression, as in Fig. 2, the parenthesis ‘据 报道(it is reported)’, which shows the source of information for the following statement, is deleted for sentence compression because it seems to be an independent element from the other parts of the sentence.

⁸ The parenthesis refers to the elements in a sentence which functions as the explanatory or qualifying remarks and has no clear dependent relations with the other constituents of a sentence. The parenthesis is usually delimited with a comma if it locates at the beginning or the end of a sentence, or two commas if it locates in the middle of a sentence.

For better understanding of the fundamental rules discussed above for annotation of Chinese sentence compression, we list them in table 2 with examples.

3.3 Quality Control

Three annotators⁹ participate in this task. To guarantee high annotation quality, we implement a two-phase process: phase 1 is a multi-round pilot annotation on small-size datasets for training and phase 2 is a formal annotation on the full size dataset.

Table 3. Statistics of three-round pilot annotation

Round	# Sentences	Compression Rate		Kappa
1	30	Human1	0.717	0.652
		Human2	0.685	
		Human3	0.657	
2	50	Human1	0.559	0.841
		Human2	0.581	
		Human3	0.566	
3	50	Human1	0.551	0.886
		Human2	0.557	
		Human3	0.535	

Table 4. Statistics of formal annotation on Parts 9-10

Dataset	# Sentences	Compression Rate		Kappa
Part9	273	Human1	0.543	0.885
		Human2	0.576	
		Human3	0.571	
Part10	290	Human1	0.512	0.860
		Human2	0.551	
		Human3	0.536	

For each round of annotation, we calculate the compression rate with each annotator's work and Fleiss' Kappa [6] for the inter-annotator agreement. After each round of annotation, the manual is revised based on our review of the inter-annotation inconsistencies and discussion about the ambiguous cases. Only after Fleiss' Kappa indicates the inter-annotator agreement is satisfactory and remains stable will the pilot annotation stop and the formal-run annotation start.

⁹ All three of the annotators are Chinese natives and have received considerable training in linguistics, particularly in syntax.

During the phase of formal annotation, two annotators work on Parts 1-8 (2,745 sentences), providing one reference result for each source sentence. To further check the inter-annotator consistency, the three annotators work on Part 9 and Part 10 respectively, and thus each sentence in these two datasets has three reference compression results.

4 Evaluation and Analysis

Using the corpus presented above, we develop a Chinese sentence compression system and study various evaluation methods for this task.

4.1 Automatic Sentence Compression

We use Tree Transducer Toolkit (T3)¹⁰ to build a Chinese sentence compression system. T3 is a tree-to-tree transduction model based on synchronous tree-substitution grammars (STSGs), which achieves state-of-the-art performance in the English sentence compression tasks [4]. To enable T3 to perform on the Chinese data, we modify the head-finding rules according to the Chinese head rules described in [1]. We use the data of Parts 1-8 as the training set, Part 9 as tuning set and Part 10 as the test set. To obtain the n -gram feature, we train a tri-gram language model on the Xinhua and AFP Portions of the GIGAWORD Chinese corpus. Since T3 requires CTB-style trees, we use the Berkeley parser¹¹ to parse all the sentences¹². By default we choose the asymmetric hamming distance loss function for the large margin training of the system.

4.2 Evaluation Metrics

Like in English sentence compression tasks, we choose grammatical relation F1 as one of the evaluation metrics, which allows us to measure the semantic aspects of summarization quality in terms of grammatical-functional information [14]. We use the ZPar dependency parser¹³[23] to extract Chinese grammatical relations for all the sentences in the test set and gold references.

In principle, the sentence compression evaluation is to compute the errors of a sentence against its gold reference(s), which is similar to the evaluation of MT systems, especially when no paraphrasing is performed in compression as we do in this work. Therefore, we adopt additional MT evaluation metrics in our experiment. Specifically we choose three n -gram and similarity-based metrics, BLEU[13], NIST[5] and GTM[17], which are very popular in automatic evaluation methods in MT. Besides, we use three Levenshtein distance based metrics, mWER[12], mPER[16], and

¹⁰ <http://staffwww.dcs.shef.ac.uk/people/T.Cohn/t3/>

¹¹ <http://code.google.com/p/berkeleyparser/downloads/list>

¹² It would be interesting to compare the results of using automatic parsers and gold parse trees. We leave it for our future work.

¹³ <http://www.sutd.edu.sg/cmsresource/faculty/yuezhang/zpar.html>

TER[15], which regard the evaluation problem as pairwise string alignment between the output string and the gold reference¹⁴.

To study the correlation between the automatic evaluation measures and human judgments, we also conduct human evaluation on the same data. Judges are required to separately rate along a 5-point scale how much information the compressed sentence retains against the source sentence (i.e., *importance*) and how grammatical the compression is without the presence of the source sentence (i.e., *grammaticality*).

Table 5. Evaluation results with different metrics and different number of references

Metrics	Ref1	Ref2	Ref3	Ref1-3
NIST	7.423	6.889	7.017	7.938
BLEU	0.521	0.508	0.498	0.641
GTM	0.694	0.700	0.685	0.739
mWER	0.516	0.497	0.520	0.445
mPER	0.471	0.457	0.459	0.407
TER	0.495	0.484	0.506	0.418
Relational F1	0.574	0.587	0.547	0.652

Table 6. Human evaluation results of different system outputs

Entry	Com.R	Importance	Grammaticality
Output 1	0.400	3.803	4.228
Output 2	0.341	3.438	4.017
Output 3	0.631	4.252	4.566
Output 4	0.424	3.700	4.183
Output 5	0.427	3.679	4.152

4.3 Results and Analysis

Table 6 shows the automatic evaluation results on the test set with single and multiple references. We see, first of all, that the results of each reference are not very stable and show irregular variance by different measures. We attribute this to the ambiguity of sentence compression tasks, that is, even though annotators can get agreement in most cases, there exists some cases with more than one correct answer. This explanation is further confirmed when we switch to the multi-reference evaluation. By the 3-reference result, all measures show significant different scores (or better performance) with the single-reference counterparts, indicating that the sentence compression task has some natural ambiguities which cannot be eliminated, even for well-trained native language annotators. Therefore, for reliable estimation of the compression system performance, it is necessary to conduct evaluation with more than one reference. This finding is actually somewhat

¹⁴ Not that, for grammatical relation F1, BLEU, NIST and GTM, larger values reflect better translation quality. For mWER, mPER and TER, smaller means better.

similar to that in machine translation where correct translations are plenty and the evaluation against a single reference is very unreliable [13].

We then examine how the automatic measures correlate with human judgments. To conduct evaluation on diverse compression results, we generate five outputs with different loss functions used in the T3 toolkit¹⁵. 7 Chinese native-language judges participate in the evaluation and score each system output by the rating schema presented in Section 4.2. Table 6 shows that compression rate is an important factor for a successful Chinese sentence compression system. For example, the best result (output 3) is achieved when the compression rate is closest to those of the references, while the worst result (output 2) corresponds with a compression rate that is the farthest from the reference rates as shown in Table 4.

Table 7. Correlation coefficients between automatic measures and human judgments

Entry	NIST	BLEU	GTM	mWER	mPER	TER	Relational F1
Importance	0.843	0.807	0.870	-0.887	-0.884	-0.718	0.896
Grammaticality	0.827	0.798	0.878	-0.904	-0.900	-0.732	0.905

Finally we plot the automatic measures as functions of the human evaluation scores¹⁶. As shown in Fig. 3, most of these measures correlate well with the human judgments on outputs 1, 2, 3 and 5. However, they cannot distinguish well between output 1 and output 4 which are quite close in human evaluation. This phenomenon reflects a limited ability of current automatic metrics in prediction on similar compression results. Furthermore, we use the Pearson Correlation Coefficients to estimate the correlation degree. Table 7 shows that relational F1 correlates best with judges (around 0.90), which agrees with the observation seen in the English tasks[3]. More interestingly, it is observed that GTM, mWER and mPER obtain very good correlation scores (absolute value > 0.87), followed by BLEU and NIST (absolute value > 0.79). These results indicate a very promising application of MT evaluation methods in Chinese sentence compression tasks.

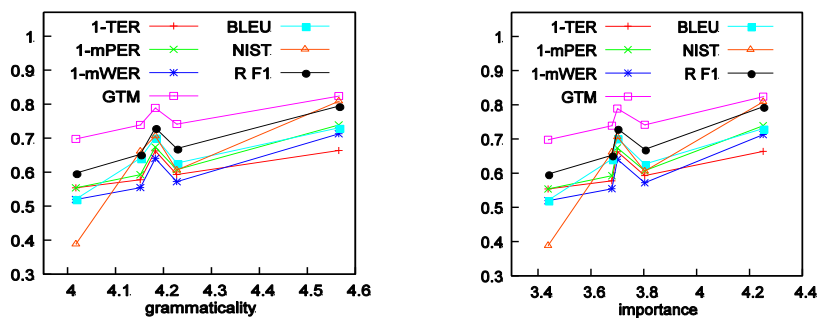


Fig. 3. Automatic measures of the sentence compression results against human judgments

¹⁵ The 5 outputs correspond with loss function 2, 8, 10, 16 and 17 respectively.

¹⁶ As mWER, mPER and TER have negative correlations with human evaluation scores, we use 1-mWER, 1-mPER and 1-TER as functions for a clear presentation. Also, the NIST score is normalized with a factor of 12 to fit it into the range of [0, 1].

5 Conclusion and Future Work

We have presented a first-ever manually-built Chinese sentence compression corpus. By using this corpus, we develop an automatic sentence compression system and study various evaluation methods on this task. We find that 1) using multiple references is necessary for automatic evaluation; and 2) besides relational F1, some MT evaluation measures are also well correlated with human judgments, and are very promising for the evaluation of sentence compression systems.

In the future we would like to enlarge this Chinese sentence compression corpus by annotating the other parts of the CTB data and apply the corpus to some NLP tasks like machine translation.

Acknowledgements. This work was supported in part by the National Science Foundation of China (61073140; 61272376), Specialized Research Fund for the Doctoral Program of Higher Education (20100042110031) and the Fundamental Research Funds for the Central Universities (N100204002).

We would like to thank Dirk Hovy for providing his kappa calculator. We also thank Yue Zhang for his assistance with the ZPar statistical parser. Finally we thank Matt Snover for assisting us to use the TER metric in our Chinese sentence compression evaluation work.

References

1. Bikel, D.M.: Intricacies of Collins' Parsing Model. *Computational Linguistics* 30(4), 479–511 (2004)
2. Clarke, J., Lapata, M.: Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research* 1(31), 399–429 (2008)
3. Clarke, J., Lapata, M.: Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In: *Proceedings of ACL-COLING*, pp. 377–384 (2006b)
4. Cohn, T., Lapata, M.: Sentence compression beyond word deletion. In: *Proceedings of the 22nd COLING*, pp. 137–144 (2009)
5. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: *Proceeding of HLT 2002 Proceedings of the Second International Conference on Human Language Technology Research*, pp. 138–145 (2002)
6. Fleiss, J.L.: Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5), 378–382 (1971)
7. Galley, M., McKeown, K.R.: Lexicalized Markov Grammars for Sentence Compression. In: *Proceedings of HLT-NAACL*, pp. 180–187 (2007)
8. Jing, H.: Sentence Reduction for automatic summarization. In: *Proceedings of ANLP*, pp. 310–315 (2000)
9. Knight, K., Marcu, D.: Statistical-based summarization-step one: sentence compression. In: *Proceedings of AAAI 2000*, pp. 703–710 (2000)
10. Knight, K., Marcu, D.: Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence* 139(1), 91–107 (2002)
11. McDonald, R.: Discriminative sentence compression with soft syntactic constraints. In: *Proceedings of EACL*, pp. 297–304 (2006)

12. Nießen, S., Och, F.J., Leusch, G., Ney, H.: An evaluation tool for machine translation: Fast evaluation for machine translation research. In: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC), pp. 39–45 (2000)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of 40th Annual Meeting of ACL, pp. 311–318 (2002)
14. Riezler, S., King, T.H., Crouch, R., Zaenen, A.: Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In: Proceedings of HLT-NAACL, pp. 118–125 (2003)
15. 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)
16. Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., Sawaf, H.: Accelerated DP based search for statistical translation. In: Proceedings of European Conference on Speech Communication and Technology, pp. 2667–2670 (1997)
17. Turian, J.P., Shen, L., Melamed, I.D.: Evaluation of Machine Translation and its Evaluation. In: Proceedings of MT Summit IX, pp. 386–393 (2003)
18. Turner, J.P., Charniak, E.: Supervised and unsupervised learning for sentence compression. In: Proceedings of 43rd Annual Meeting of ACL, pp. 290–297 (2005)
19. Xue, N., Xia, F., Chiou, F., Palmer, M.: The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering* 11(2), 207–238 (2005)
20. Yamangil, E., Nelken, R.: Mining Wikipedia revision histories for improving sentence compression. In: Proceedings of 46th Annual Meeting of ACL, pp. 137–140 (2008)
21. Yamangil, E., Shieber, S.M.: Bayesian synchronous tree-substitution grammar induction and its application to sentence compression. In: Proceedings of 48th Annual Meeting of ACL, pp. 934–947 (2010)
22. Yoshikawa, K., Iida, R., Hirao, T., Okumura, M.: Sentence Compression with Semantic Role Constraints. In: Proceedings of 50th Annual Meeting of ACL, pp. 349–353 (2012)
23. Zhang, Y., Clark, S.: Syntactic Processing Using the Generalized Perceptron and Beam Search. *Computational Linguistics* 37(1), 105–151 (2011)

Graphic Language Model for Agglutinative Languages: Uyghur as Study Case

Miliwan Xuehelaiti^{1,2,3,*}, Kai Liu², Wenbin Jiang², and Tuergen Yibulayin¹

¹ Xinjiang University Information Science and Technology Institute
Urumqi, Xinjiang 830046, China
`mihreban@126.com`

² Institute of Computing Technology, University of Chinese Academy of Sciences
Beijing 100190, China
`{liukai, jiangwenbin}@ict.ac.cn`

³ Urumqi Administration of Industry and Commerce
Urumqi, Xinjiang 830002, China

Abstract. This paper describes a novel, graphic language modeling strategy for morphologically rich agglutinative languages. Different from the linear structure in n-gram language models, graphic modeling organizes the morphemes in a sentence, including stems and affixes, as a directed graph. The graphic language model is verified in two typical application scenarios, morphological analysis and machine translation. We take Uyghur for example, and experiments show that the graphic language model achieves significant improvement in both morphological analysis and machine translation.

Keywords: graphic language model, agglutinative language, morphological analyzing, statistical machine translation.

1 Introduction

Language model is one of the most important models in NLP, it describes probabilities of sentences in natural language. Many NLP tasks can be boiled down to the modeling of language model, such as transliteration, speech recognition, part of speech tagging and so on. One of most widely used language model is n-gram language model[3], which models words in sentences with local context environment in an linear way. The n-gram language model is simple and effective, and it have got excellent performance on Chinese, English and other languages with simple morphological form.

Agglutinative language is one kind of language which is widely used in North/South Korea, Japan, Mongolian, Turkey and other countries in Middle East Asia and other areas. Agglutinative languages differ from languages with simple morphological form (such as English and Chinese) in their sentence and

* This work is supported by the National Science Foundation of China (Grant No. 61262060 61262060), Key Project of National Natural Science Fund [61032008] and National Social Science Fund Key Projects [10AYY006].

word-formation[2], in which the composition of each word in agglutinative language follows different word-building rules according to simple observation: each word of agglutinative language is composed by a word-stem and any number of affixes, in where constrained relations exist between stem and affixes; and similar relations exist in stems of different words. The former rule lead to the data sparseness of word of agglutinative language, the latter rule makes it hard to seize the relation between stems, for there maybe some affixes between stems in different words.

According to the observations above, sentence of agglutinative language with those relations can not be simply modeled as linear sequence. As a matter of fact, traditional n-gram language language model which models sentences as linear sequence can not obtain idea results on agglutinative languages. In this paper, we propose a novel graphic language model which can depict those relations more deeply. More specifically, our graphic language model models the generative relations between stem and affixes in a word and the relations between stems in different words, which relations can hardly be modeled by traditional linear language models.

In order to test the novel graphic language model, two language model needed natural language processing tasks (morphological analyzing and statistical machine translation(SMT)) are adopted to verify our graphic language model. In the experiments, both tasks show that graphic language model gets significant improvements compares to n-gram language model. In morphological analyzing, the accuracy gains 0.8% improvements due to the new style language model, while in SMT it gains more than 1.1 BLEU improvement. Furthermore, the graphic language model is simple, and the complexity of it is approximate to the n-gram language model.

The rest of paper is organized as follows: Section 2 describes the characteristics of agglutinative languages; Traditional linear language model is described in Section 3; We propose our graphic language model for agglutinative language in Section 4; And the methods of utilizing proposed graphic language model in two NLP tasks are shown in Section 5; Finally, we present the experiments of the two NLP tasks with graphic language model on Uyghur in Section 6 and conclude in Section 8.

2 Agglutinative Language

Agglutinative language is a kind of language that its words are made up of distinct morphemes by a linear sequence way, and each component of meaning is represented by its own morpheme. Agglutinative languages have many characteristic, more specifically, we take Uyghur as our study case. Uyghur is one of typical agglutinative languages, it is a Turkic language which is widely used in Western China by Uyghur people, and it shares some characteristics with other agglutinative languages:

Table 1. An example of agglutinative language’s word with multiple morphemes. A word with different morphemes will show different meanings and even be a short sentence.

Word	Stemming	Meaning
<i>Ölchem</i>	<i>Ölchem</i>	standard
<i>Ölchemlesh</i>	<i>Ölchem+lesh</i>	standardization
<i>Ölchemleshtür</i>	<i>Ölchem+lesh+ür</i>	standardize (it)
<i>Ölchemleshtürel</i>	<i>Ölchem+lesh+ür+el</i>	can standardize (it)
<i>Ölchemleshtürelme</i>	<i>Ölchem+lesh+ür+el+me</i>	can not standardize (it)
<i>Ölchemleshtürelme</i>	<i>Ölchem+lesh+ür+el+me+m</i>	can not standardize (it)?
<i>Ölchemleshtürelmesiler</i>	<i>Ölchem+lesh+ür+el+me+m+siler</i>	can’t you standardize (it)?

- Each word jointed with different affixes will show different meanings. Take a word in Uyghur as example, the word ”xizmet”(job) will show different meanings when different morphemes followed with it: ”xizmettin”(from work), ”xizmetde”(with work) and so on.
- Each word can be jointed with multiple morphemes, and such a word can even be a short sentence. As it is shown in Table 1, the same word ”*Ölchem*”(standard) jointed with different morphemes convey different meanings. And much important information, such as, meanings of content words are conveyed by those morphemes.

In addition, morphemes of Uyghur fall into two categories: stem and affix. The first morpheme of each word is the stem of the word in Uyghur, and each word should have one and only one stem, which conveys the main semantic meaning of the word. As the example above, ”xizmet”(job) and ”*Ölchem*”(standard) are stems. And all morphemes after stems are affixes, which convey minor semantic meanings or grammar information. Furthermore, all stems in a sentence form the skeleton of the sentence.

3 Linear Language Model

Language model describes a word sequence w_j^i ($w_i, w_{i+1}, \dots, w_{j-1}, w_j$) by assigning the probability $P(w_j^i)$ to the sequence by means of certain probability distribution. And language model is widely used in many NLP applications, such as speech recognizing, morphological analyzing, machine translation and so on.

Most language models regard the word sequence as a linear structure, and calculate the probability in a linear way. One of the most typical models is n-gram language model, which assigns a given word’s probability by means of its previous contiguous words. In n-gram language model, the probability of the sequence w_j^i is assigned approximated as:

$$P(w_j^i) = \prod_k P(w_k | w_{k-1}^i) \approx \prod_k P(w_k | w_{k-1}^{k-n+1}) \quad (1)$$

Table 2. The lexical statistical data for Uyghur and Chinese. Total word: the total number of words existed in the corpus; total freq: the sum of all words' frequencies; avg freq: the average frequency of each word.

	Uyghur	Chinese
total word	69563	43285
total freq	1263861	1161801
avg freq	18.17	26.84

where n is the n -gram size of the language model. And $P(w_k|w_{k-1}^{k-n+1})$ can be calculate from its frequency in corpus:

$$P(w_k|w_{k-1}^{k-n+1}) = \frac{\#(w_k^{k-n+1})}{\#(w_{k-1}^{k-n+1})} \quad (2)$$

where $\#(\cdot)$ means the total count of the n -gram in the corpus. And in practical terms, the probability above needs some kind of smoothing, such as "add-one", Good-Turing, Kneser-Ney smoothing and so on[4].

3.1 Linear Modeling for Agglutinative Language

Because of the characteristic of agglutinative language, common linear modeling for agglutinative language will encounter some problems:

- Data sparseness. For each word in agglutinative language can be made of several morphemes, and the number of all probable words are astronomical, which can make serious data sparseness problem. Take Uyghur and Chinese as example, we count the lexical frequencies of both languages in parallel corpus (Table 2), from where we can see less frequency per word in Uyghur, in other words, more data sparseness.
- Ignoring the relations in/between words. Traditional linear language model ignore to model relations between stems and relations inside the word. And a remedial measure for it is to model morphemes as basic units instead of words in n -gram model. This measure can describe stem-affix relations, but it still have weaker ability to describe the relations between stems.

4 Graphic Language Model

Since there are several drawbacks of linear modeling agglutinative language, we propose a morpheme based directed graphic language model. And the directed graphic structure can better describe the characteristic of agglutinative language.

As it is shown in Figure 1, we model a agglutinative language sentence as directed graphic with morphemes as basic elements, where all stems are connected linearly in left-right order and all affixes are connected to previous affixes or stems. And it can be divided into two linear parts:

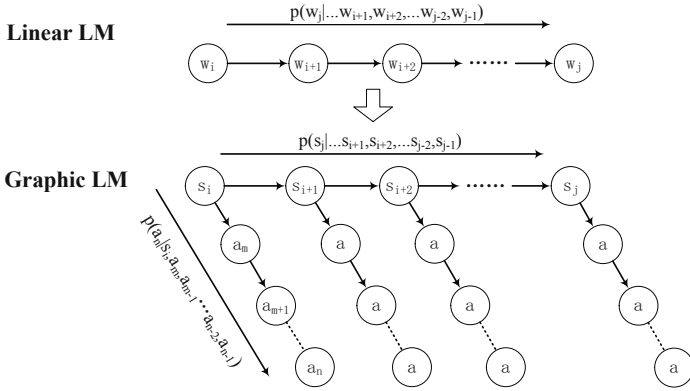


Fig. 1. The structures of linear language model (Linear LM) and graphic language model (Graphic LM). Compared to linear language model, graphic language model describe the detail relation of stems and affixes inside/outside the words. And as it is shown in the figure, it can be separated into two parts of sub-models(stem-stem and stem-affix).

- stem-stem: this part assigns the probability to all stems, similar to n-gram model, the probability can be calculated approximately by assuming the probability of observing stem can be calculated in condition of preceding n stems:

$$P(s_i^1) = \prod_i P(s_i | s_{i-1}^1) \approx \prod_i P(s_i | s_{i-1}^{i-n+1}) \tag{3}$$

where s_i denotes the stem of i^{th} word.

- stem-affix: this part calculates all affixes probabilities by means of preceding stem and affixes in the same word:

$$P(a) = \prod_i \prod_j P(a_{i,j} | s_i, a_{i,j-1}^{i,1}) \tag{4}$$

where $a_{i,j}$ denotes the j^{th} affix in i^{th} word in the sentence.

For the whole model, we combine both parts above together, and calculate sentences' probabilities as follows:

$$P(w_n^1) = \prod_i (P(s_i | s_{i-1}^{i-n+1}) \prod_j P(a_{i,j} | s_i, a_{i,j-1}^{i,1})) \tag{5}$$

where stems s and affixes a are obtained from morphological analyzing results of the sentence w_n^1 . The training method of both parts of this model can refer to n-gram language model. In this way, we design a morpheme based directed graphic language model, which is supposed to be a better language model for agglutinative language.

Algorithm 1. Estimating Probability of Sequence

```

1: Stemmed Seq ← MA(Seq)
2: Stem-Stem Seq ← RemoveAffix(Stemmed Seq)
3: Stem-Affix Seq List ← Split(Stemmed Seq)
4: Stem-Stem Prob ← LM(Stem-Stem Seq, Stem-Stem Model)
5: for each Stem-Affix Seq ∈ Stem-Affix Seq List do
6:   Stem-Affix Prob ← LM(Stem-Affix Seq, Stem-Affix Model)
7: end for
8: Graphic Prob ← Multiply(Stem-Stem Prob, Stem-Affix Prob)

```

Training. Firstly, we morphological analyze the training corpus into stemmed one. Secondly, according to the stemmed corpus, we obtain stem-stem and stem-affix corpora by removing affixes and splitting sentences by words respectively. Then, those corpora are utilized to train linear language model for stem-stem and stem-affix respectively.

Estimating Probability. Algorithm 1 outlines the estimation procedure in its entirety. In line 1, we analyze the input sequence by morphological analyzing procedure $MA(\cdot)$ and obtain the stemmed corpus. The stemmed corpus is utilized to obtain stem-stem sequences by removing all affixes in the corpus in line 2. Correspondingly, in line 3 sentences are split according to words and then organized into stem-affix sequences. From line 4-7, we calculate the both parts' probabilities with corresponding sequences and sub models through procedure $LM(\cdot, \cdot)$. And the final score of the model is combined by scores from sub-models in line 8.

5 Applications

5.1 Morphological Analyzing

Morphological analyzing is one of the most important NLP tasks in agglutinative language[1]. The quality of morphological analyzing will affect other NLP tasks, which are based on morphological analyzing. There are three sub-task in morphological analyzing, including stemming, restoring the changed letter and POS tagging, in which we select the first sub-task stemming as the application to verify our graphic language model. Stemming is similar to segmentation, it splits each word into morphemes (Figure 2), including a stem and several following affixes. According to the characteristic of agglutinative language, stemming needs the contexts of inside or outside the word, where language model is available and important.

Formally, we define a word sequence as w_j^i , which means it is a word sequence with words from position i to j in sentence. And each word can be segmented into several morphemes m , which contain a single stem s and several following affixes a . In this task, we try to find the most probable morphological segmentation m_n^1 of sentence w_n^1 .

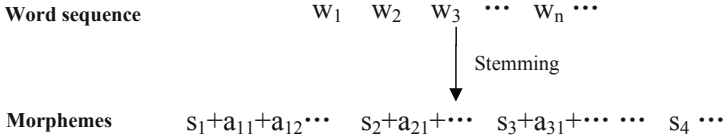


Fig. 2. Stemming word sequence into morphemes, where the first morpheme in each word is stem and others are affixes

With linear modeling, n-gram language model selects the morpheme sequence with the maximum language model probability:

$$\ell(w_n^1) = \arg \max_{m_{n'}^1} \prod_i P(m_i | m_{i-1}^{i-n+1}) \tag{6}$$

where m denotes a morpheme, and $m_{n'}^1$ is the selected morpheme sequence of the sentence w_n^1 . And the $P(m_i | m_{i-1}^{i-n+1})$ means n-gram language model’s probability of morpheme m_i with context m_{i-1}^{i-n+1} .

With graphic modeling, correspondingly, we try to find the morpheme sequence with the maximum model probability with stems and affixes:

$$\ell(w_n^1) = \arg \max_{s,a} \prod_i P(s_i | s_{i-1}^0) \prod_{i,j} P(a_{i,j} | s_i, a_{i,j-1}^{i,0}) \tag{7}$$

where s_i denotes the stem of the i^{th} word, and $a_{i,j}$ denotes the j^{th} affix of i^{th} word. The first term of Formula 7 is the stem-stem part of our graphic model and the second term is the stem-affix part.

5.2 Machine Translation

Machine translation is one of the hardest problems in NLP. The performance of statistical machine translation is highly depended on the quality of the language model (cite), and it is a good task to verify the quality of language model. In this paper, we try to verify the effectiveness of graphic language model with agglutinative language as target side.

Due to the characteristics of agglutinative language, the application will be performed on two different SMT system with different granularities:

Word Based. The SMT model is trained with words in agglutinative language side as the basic translation unit, and this kind of SMT system has several characteristics:

- It has large-grained translation unit, which may suffer from data sparseness problem.
- Shorter sequences of agglutinative language will be translated while we use word as basic translation unit.
- Word based translation system is free to recombination of morphemes into words.

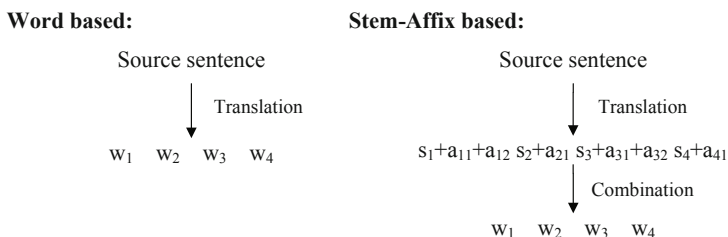


Fig. 3. Translation from one language to an agglutinative language with different basic units. Left panel shows the translation from source language directly to target words, while the right panel shows the translation procedure that translate source language firstly into morphemes and then combine them into final words.

Morpheme Based. Correspondingly, the stemmed sentences are used to train the SMT model and stems and affixes are the basic translation unit here:

- Smaller-grained translation unit means less data sparseness problem.
- Longer sequences have to be translated while stems and affixes are chosen, which means higher translation complexity.
- Morpheme based translation system have to recombine stems and affixes into agglutinative words.

6 Experiments

In this section, we verify our graphic language model through two applications. And there are three different types of language model will be utilized in the experiments:

- Word linear LM: the n-gram language model based on words, and it will be utilized in the application of SMT.
- Morphemes linear LM: morpheme based n-gram language model which will be utilized in both applications.
- Graphic LM: our graphic language model for agglutinative language, and the order (n) of linear part is equal to corresponding comparison LM in experiments as default.

6.1 Morphological Analyzing

DataSet. We make use of an annotated corpus Mega-words Corpus of Morphological Analysis of Uyghur, which is manually annotated by Xinjiang multilingual key laboratory. And it contains about 67 thousands sentences, from which we select 5% as our testing set.

Table 3. Experiment results on morphological analyzing with different language model

Model	P%	R%	F%
Morpheme linear LM	87.5	87.4	87.5
Graphic LM	88.0	88.6	88.3

Training and Evaluation. We train a 5-gram language model on morphemes and the linear part of our graphic model by SRI Language Modeling Toolkit[9] with Kneser-Ney smoothing. And we simply evaluate the stemming result by precision and recall of the morphemes.

Results. As the results shown in Table 3, our graphic language model shows advantage on morphological analyzing compared to morpheme linear language model, where precision obtain an improvement of 0.5% and more than 1% improvement on recall.

6.2 Machine Translation

In this section, we compare our graphic language model with linear n-gram language model in SMT task. In SMT task, two different granularities are employed to verify the effectiveness of our graphic language model. One experiment is performed on word, while the other is performed on the results of morphological analyzing (stems and affixes).

DataSets. For bilingual training data, we select Chinese-Uyghur corpus with 120 thousand parallel sentence pairs, which includes fifty thousand sentence pairs from corpus provided by CWMT 2011 evaluation task[5]. We obtain morphological result of the corpus by performing Uyghur morphological analyzer¹ on the corpus. The parallel corpus’s word alignments are obtained by running GIZA++[6] on the corpus in both directions and applying ”grow-diag-and” refinement.

Training and Evaluation. We use the development set provided by CWMT 2011² evaluation task as our development set, and we organize 1000 sentence pairs as our own test set. The quality of translation is evaluated by the NIST BLEU-4 metric[7]. We make use of the standard MERT as the tuning algorithm to tune our cascaded translation model’s parameters on development set.

Baselines and Our Model. We apply SRI Language Modeling Toolkit to train language models with modified Kneser-Ney smoothing on Uyghur side of the training corpus. The open source SMT decoder Moses[8] is selected as our baseline, which contains implementation of hierarchical phase model (Moses-chart). Correspondingly, our model is based on the same decoding system Moses, and train our graphic language model on the same corpus (training corpus).

¹ Developed by Institute of Computing Technology(ICT), Chinese Academy of Sciences(CAS)

² http://www.chineseldc.org/resource_info.php?rid=156

Table 4. Experiment results on test set. We test translation model with different language model respectively. word: word based n-gram language model; stem: stem part of our graphic language model; affix: affix part of our graphic language model; morpheme: the morpheme based linear language model. And stem+affix is our whole graphic language model. The followed number denotes the order of the language model, for example, word5 means a 5-gram word based language model and stem3 means we train the stem part with order 3.

Granularity	Language Model	BLEU%
Word based	word5	51.19
	word5+morpheme5	52.28
	word5+stem3	53.01 (+0.73)
	word5+stem5	53.18 (+0.90)
	stem3+affix3	53.18 (+0.90)
	stem5+affix5	53.44 (+1.16)
Morpheme based	morpheme5	54.26
	stem3+affix3	54.91 (+0.65)
	stem5+affix5	55.26 (+1.00)

Results. The experiment result is shown in Table 4, which line 2-7 show the results of word based SMT model with different language model respectively. And line 9-10 give the results of morpheme based SMT model with both linear language model and our graphic language model.

As the results shown, our graphic language model is significant better than those linear modeling language model. And both parts of our graphic language model (stem-stem, stem-affix) show their effectiveness on experiment, while the whole model shows better performance. Meanwhile, those improvements prove that it is reasonable to model agglutinative language in stem-stem and stem-affix style, and this style of structure can describe some kinds characteristic of agglutinative language.

7 Related Work

Language Model. In addition to n-gram language model, there are much work is devoted into language model. Some kind of structured language model aims at modeling the structures of language and overcoming the locality problem[10] and neural network is employed to improve the work[11]. But so far, there is not any work on the structure of agglutinative language.

Morphological Analyzing. There are a lot of supervised work on morphological analyzing for each language respectively: Japanese[12], Arabic[13], and so on. Correspondingly, unsupervised ones (e.g.[14]) are also available. And morphological analyzing is proved to be an important task for other NLP task (e.g. SMT [16–18]).

Machine Translation. So far, most studies of agglutinative related machine translation are base on agglutinative to non-agglutinative translation, such as, for Turkish[15–17], Korean[18, 19] and others[20]. And there is also work on alignment between agglutinative language and other languages in translation purpose[21, 22]. While there is less work on translation of non-agglutinative language to agglutinative language[23].

8 Conclusion and Future Work

In this paper, we model agglutinative language with graphic structure on the basis of the characteristics of agglutinative language by observations. The novel language model can better describe the agglutinative language and remit data sparseness, where evidences are provided by the experiments of different NLP tasks. The experiment results show significant improvements on morphological analyzing and SMT tasks with 0.8 F-score improvements and 1.1 BLEU improvements.

For future work, we will investigate other kinds of structures that can better model the agglutinative language (e.g. indirected graph or all connected graph) and involve more feature of agglutinative language into our model, or directly model the language model as discriminative model.

References

1. Bisazza, A., Federico, M.: Morphological pre-processing for Turkish to English statistical machine translation. In: Proceedings of IWSLT, pp. 129–135 (2009)
2. Oflazer, K.: Two-level description of Turkish morphology. *Literary and Linguistic Computing* 9(2), 137–148 (1994)
3. Katz, S.: Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on, Acoustics, Speech and Signal Processing* 35(3), 400–401 (1987)
4. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, pp. 310–318. Association for Computational Linguistics (1996)
5. Zhao, H., Lü, Y., Ben, G., Huang, Y., Liu, Q.: The evaluation report of CWMT 2011. CWMT 2011, pp. 261–180 (2011)
6. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 19–51 (2003)
7. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)
8. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Annual Meeting-Association for Computational Linguistics, vol. 45, p. 2 (2007)

9. Stolcke, A., et al.: Srlm-an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing, vol. 2, pp. 901–904 (2002)
10. Chelba, C., Jelinek, F.: Structured language modeling. *Computer Speech & Language* 14(4), 283–332 (2000)
11. Emami, A., Jelinek, F.: A neural syntactic language model. *Machine Learning* 60(1-3), 195–227 (2005)
12. Nagata, M.: A stochastic Japanese morphological analyzer using a forward-DP backward-A* N-best search algorithm. In: Proceedings of the 15th Conference on Computational Linguistics, vol. 1, pp. 201–207. Association for Computational Linguistics (1994)
13. Buckwalter, T.: Buckwalter Arabic Morphological Analyzer Version 1.0 (2002)
14. Creutz, M., Lagus, K.: Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Helsinki University of Technology (2005)
15. Bisazza, A., Federico, M.: Morphological pre-processing for Turkish to English statistical machine translation. In: Proceedings of IWSLT, pp. 129–135 (2009)
16. Goldwater, S., McClosky, D.: Improving statistical MT through morphological analysis. In: Proceedings of HLT EMNLP, pp. 676–683 (2005)
17. Mermer, C., Saraclar, M.: Unsupervised Turkish morphological segmentation for statistical machine translation. In: Workshop of MT and Morphologically-rich Languages (2011)
18. Lee, Y.-S.: Morphological analysis for statistical machine translation. In: Proceedings of HLT NAACL, Short Papers, pp. 57–60 (2004)
19. Luong, M.-T., Nakov, P., Kan, M.-Y.: A hybrid morpheme-word representation for machine translation of morphologically rich languages. In: Proceedings of EMNLP, pp. 148–157 (2010)
20. Virpioja, S., Vayrynen, J.J., Creutz, M., Sadeniemi, M.: Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In: Proceedings of MT SUMMIT, pp. 491–498 (2007)
21. Luong, M.-T., Kan, M.-Y.: Enhancing morphological alignment for translating highly inflected languages. In: Proceedings of COLING, pp. 743–751 (2010)
22. Wang, Z., Lu, Y., Liu, Q.: Multi-granularity word alignment and decoding for agglutinative language translation. In: Proceedings of MT SUMMIT, pp. 360–367 (2011)
23. Yeniterzi, R., Oflazer, K.: Syntaxto-morphology mapping in factored phrase-based statistical machine translation from English to Turkish. In: Proceedings of ACL, pp. 454–464 (2010)

*i*CPE: A Hybrid Data Selection Model for SMT Domain Adaptation

Longyue Wang, Derek F. Wong, Lidia S. Chao, Yi Lu, and Junwen Xing

Natural Language Processing & Portuguese-Chinese Machine Translation Laboratory
Department of Computer and Information Science
University of Macau, Macau S.A.R., China
{mb15505, derekfw, lidiasc, mb25435, mb15470}@umac.mo

Abstract. Data selection is a significant technique to enhance the data-driven models especially for large-scale natural language processing (NLP). Recent research on statistical machine translation (SMT) domain adaptation focuses on the usage of various individual data selection models. In this paper, we proposed a hybrid data selection model named *i*CPE, which combines three state-of-the-art similarity metrics: Cosine *tf-idf*, Perplexity and Edit distance at both corpus level and model level. We conduct the experiments on Hong Kong Law Chinese-English corpus and the results show that this simple and effective hybrid model performs better over the baseline system trained on entire data as well as the best rival method. This consistently boosting performance of the proposed approach has a profound implication for mining very large corpora in a computationally-limited environment.

Keywords: Data Selection, Statistical Machine Translation, Domain Adaptation, Hybrid Model, Similarity Metrics.

1 Introduction

The performance of SMT [1] system depends heavily upon the quantity of training data as well as the domain-specificity of the test data with respect to the training data. A well-known challenge is that the data-driven system is not guaranteed to perform optimally if the data for training and testing are not identically distributed. Thus, domain adaptation techniques are employed to improve the translation quality for a text in a particular domain using some mixture of in-domain and out-of-domain data.

Researchers discussed the domain adaptation problems for SMT in various perspectives such as mining unknown words from comparable corpora [2], weighted phrase extraction [3], corpus weighting [4] and mixing multiple models [5, 6, 7], etc. Actually, data selection is one of the corpus weighting methods¹ [8]. One of the dominant approaches is to select data suitable for the target domain from a large general-domain corpus (general corpus). There is an underlying assumption that the general corpus is broad enough to cover a certain amount of sentences that fall in target domain². A domain-adapted machine

¹ They are data selection, data weighting and translation model adaptation.

² It is also defined as *pseudo in-domain subcorpus* by Axelrod et al. [12].

translation system could then be trained on these subcorpora instead of the entire general corpus. These supplementary data selection approaches play an important role in i) improving the quality of word alignment, ii) preventing irrelevant phrase pairs, and iii) optimizing the re-ordering of output sentences.

Until now, three state-of-the-art selection criteria have been discussed in different perspectives. The first is cosine *tf-idf* (term frequency-inverse document frequency) similarity. Hildebrand et al. [9] applied this technique to construct TM and LM adaptation and they show it is possible to adapt TMs for SMT by selecting similar sentences from general corpus. Furthermore, Lü et al. [10] proposed re-sampling and re-weighting methods for online and offline TM optimization, which are closer to a real-life SMT system. However, they obtained a slight improvement still using a large subset of total data. The second one is perplexity-based approaches, which is used to score text segments according to an in-domain LM. Recently, Moore and Lewis [11] derived the difference of the cross-entropy from a simple variant of Bayes rule. However, this is a preliminary study which did not yet show an improvement for MT task. It was further developed by Axelrod et al. [12] for SMT domain adaptation. The experimental results show that the fast and simple technique allows to discard over 99% of the general corpus resulted in an increase of 1.8 BLEU points. However, the improvement is not stable due to the selection threshold, which is hard to be estimated to ensure optimal translation quality. The third model is not explicitly used for SMT, but is still applicable to our scenario. Edit distance (ED) is a widely used similarity measure for example-based MT (EBMT), known as Levenshtein distance (LD) [13]. Koehn and Senellart [14] applied this method for convergence of translation memory (TM) and SMT. Then Leveling et al. [15] investigated different approaches (e.g., LD and standard IR) to find similar sentences for EBMT. Therefore, we consider edit distance as a new similarity metric for this domain adaptation task.

The analysis shows that each individual retrieval model has its own advantages and disadvantages, which result in their performance either unclear or unstable. Instead of exploring any single individual model, this paper provides a novel method to obtain a robust and effective data selection model for domain adaptation. We propose a hybrid model by performing linear interpolation on the three presented similarity metrics. We design it for both TM adaptation and LM adaptation at two levels: i) *corpus level* where joining the sub-corpora obtained via different individual model; and ii) *model level* where interpolating multiple TMs or LMs together. To compare the proposed model with the presented individual models, we conduct comparative experiments on a large Chinese-English general corpus to adapt to in-domain sentences on Hong Kong law. Using BLEU [16] as an evaluation metric, results indicate that the proposed approach can achieve consistent and significant improvement over baseline systems as well as any signal individual model.

This paper is organized as follows. We firstly review the related work in Section 2. The proposed and other related similarity models are described in Section 3. Section 4 details the configurations of experiments. Finally, we compare and discuss the results in Section 5 followed by the conclusions to end the paper.

2 Background

In this section, we revisit three state-of-the-art data selection models: cosine *tf-idf*, perplexity and edit distance.

2.1 Cosine *tf-idf*

Cosine *tf-idf* similarity metric comes from the realm of information retrieval (IR). It is a simple but effective co-occurrence (e.g., word overlap) based matching, which is calculated by

$$w_{ij} = tf_{ij} \times \log(idf_j) \quad (1)$$

in which each document D_i is represented as a vector $(w_{i1}, w_{i2}, \dots, w_{in})$, and n is the size of the vocabulary. tf_{ij} is term frequency (TF) of the j -th word in the vocabulary in the document D_i , and idf_j is the inverse document frequency (IDF) of the j -th word calculated. The similarity between two texts is then defined as the cosine of the angle between two vectors [17, 18]. It is good at retrieving similar (genres) sentences as well as reducing the number of out-of-vocabulary (OOV) words. However, only considering individual keyword may result in weakness in filtering irrelevant data (noises).

In practice, we only use the sentences in source language for indexing and query generating. Each sentence in general corpus is indexed as one document by Apache Lucene³. And each sentence without stop words from the reference set is used as one separate query. Besides, we make use of duplicated sentences which is similar with [9]. All retrieved sentences with corresponding target parts are ranked according to their similarity scores. Supposed that M is the size of query set and N is the number of sentences retrieved from general corpus according to each query. Thus, the size of the new sub-corpus is $Size_{Cos-IR} = M \times N$.

2.2 Perplexity Based

Perplexity can be found in the field of language modeling. As similarity metrics, it employs an n -gram language model, which considers not only the distribution of terms but also the collocation. Perplexity PP and cross-entropy $H(x)$ are monotonically related and $H(x)$ is often applied as a cosmetic substitute of PP [11]. Until now, there have been three perplexity-based variants explored for SMT domain adaptation. Among them, a metric that sums cross-entropy difference over both sides shows the best performance for this topic [12]. Cross-entropy difference is helpful to select the sentences that are more similar to in-domain corpus but different from others in general corpus. Besides, considering the bilingual resources are useful in balancing the OOV and noises. However, its performance is very sensitive to quality and quantity of the model trained on a reference set. This bilingual cross-entropy difference can be simply represented as follows:

³ Available at <http://lucene.apache.org>

$$[H_{I-src}(x) - H_{G-src}(x)] + [H_{I-tgt}(x) - H_{G-tgt}(x)] \quad (2)$$

where $H_I(x)$ and $H_O(x)$ are the cross-entropy of string x according to a language model LM_I and LM_O which are respectively trained by in-domain data set I and general-domain data set G . src and tgt are the source and target side of training data.

The candidates with lower scores have higher relation to in-domain set. The size of the new subset $Size_{PP}$ should be equal to $Size_{Cos-IR}$. Besides, we perform SRILM toolkit⁴ [19] to conduct 5-gram LMs with interpolated modified Kneser-Ney discounting [20].

2.3 Edit Distance Based

Edit distance based metric is much stricter than the former two methods, because words overlap, order and position are all involved in similarity calculation. This seems to be able to find the most ideal sentences. Given a sentence s_G from general corpus and a sentence s_R from the reference set, the edit distance for these two sequences is defined as the minimum number of edits, i.e. symbol insertions, deletions and substitutions, needed to transform s_G into s_R . Based on Levenshtein distance or edit distance, there are several different implementations. We used the normalized Levenshtein similarity score (fuzzy matching score, FMS):

$$FMS = 1 - \frac{LD(s_G, s_R)}{\text{Max}(|s_G|, |s_R|)} \quad (3)$$

which has been presented by Koehn and Senellart [14] and Leveling et al. [15]. In our system, we employed a word-based Levenshtein edit distance function. If there is a sentence of which score exceed a threshold, we would further penalize these sentences according to space and punctuations edit differences. We implemented the algorithm with map reduce technique to overcome the time-consuming problem [21].

3 The Proposed Approach

The existing domain adaptation methods can be summarized into two broad categories: i) *corpus level* by selecting, joining, or weighting the datasets upon which the models are trained; and ii) *model level* by combining multiple models together in a weighted manner [12].

For corpus level combination, we weight the sub-corpora retrieved by different methods by modifying the frequencies of the sentence in GIZA++ file [10] and then join them together. It can be formally stated as follows:

$$\begin{aligned} iCPE_{(S_x, T_x)} = & \alpha \text{CosIR}(S_x, T_x) \\ & + \beta \text{PPBased}(S_y, T_y) \\ & + \lambda \text{EDBased}(S_z, T_z) \end{aligned} \quad (4)$$

⁴ Available at <http://www.speech.sri.com/projects/srilm/>

where α , β and λ are the weights for different criteria. (S_x, T_x) , (S_y, T_y) and (S_z, T_z) are the sentence pairs respectively selected by cosine *tf-idf* (*CosIR*), perplexity-based (*PPBased*) and edit-distance based (*EDBased*).

For model level combination, we perform linear interpolation on the models trained with the sub-corpora retrieved by different data selection methods. The phrase translation probability $\phi(\bar{f}|\bar{e})$ and the lexical weight $p_w(\bar{f}|\bar{e}, a)$ are estimated using Eq. 5 and Eq. 6, respectively.

$$\phi(\bar{f}|\bar{e}) = \sum_{i=0}^n \alpha_i \phi_i(\bar{f}|\bar{e}) \quad (5)$$

$$p_w(\bar{f}|\bar{e}, a) = \sum_{i=0}^n \beta_i p_{w,i}(\bar{f}|\bar{e}, a) \quad (6)$$

where $i = 1, 2, 3$ denote phrase translation probability and lexical weight trained with the sub-corpora retrieved by *CosIR*, *PPBased* and *EDBased*. α_i and β_i are the interpolation weights.

4 Experimental Setup

4.1 Corpora

Two corpora are needed for the adaptation task. Our general-domain corpus includes more than 1 million parallel sentences comprising various genres such as newswires (LDC2005T10), sample sentences from dictionaries, law literature and other crawled sentences. The distribution of domains and sentence length of the general corpus are shown in Table 1 and Fig. 1, respectively. The in-domain corpus and test set are randomly selected that are disjointed from the LDC corpus (LDC2004T08), consisting of texts of Hong Kong law. All of them were segmented (with the same segmentation scheme)⁵ [22, 23] and tokenized⁶ [24]. In the preprocessing, we also removed the sentences with length more than 80. To evaluate the methods for both LM and TM, we used the target side sentences of the corpora to train all the LMs for translation. The sizes of the test set, in-domain corpus and general corpus we used are summarized in Table 2.

In previous work, cosine *tf-idf* method often selected data using test set as reference set [9, 10], which limits the practical applicability of the method in a real-life SMT system. For perplexity-based approaches, an in-domain corpus which is identical to the test sentences is employed for data selection [11, 12]. To compare the different methods fairly, we propose two strategies: one is *offline strategy* where we use test set to find similar sentences in general corpus; the other one is called *online strategy* where an additional in-domain corpus is used to select useful data.

⁵ IC-TCLAS2013 is available at <http://ictclas.nlpir.org/>

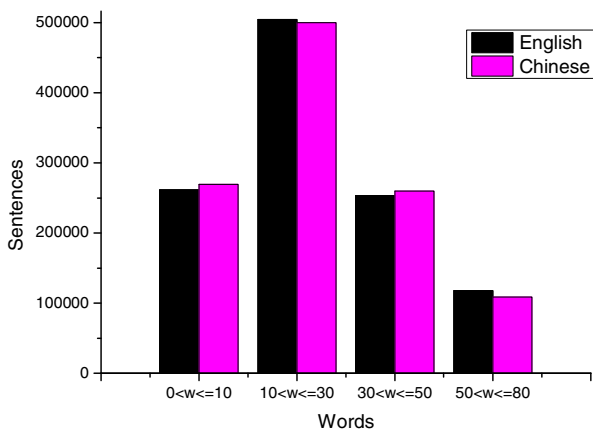
⁶ The scripts are available at <http://www.statmt.org/europarl/>

Table 1. Domain proportions in general corpus

Statistics	Domains				Total
	News	Novel	Law ^b	Miscellaneous ^a	
Sentence Number (#)	279,962	304,932	48,754	504,396	1,138,044
Percentage (%)	24.60	26.79	4.28	44.33	100.00

^a Miscellaneous part includes crawled sentences from various sources.

^b The law part includes the articles of law in Chinese mainland, Hong Kong and Macau.

**Fig. 1.** Distributions of sentences (length) of general corpus**Table 2.** Statistics summary of used corpora

Data Set	Language	Sentences	Tokens	Ave. Len.
Test Set	English	2,050	60,399	29.46
	Chinese		59,628	29.09
In-domain	English	45,621	1,330,464	29.16
	Chinese		1,321,655	28.97
Training Set	English	1,138,044	28,626,367	25.15
	Chinese		28,239,747	24.81

4.2 System Description

The experiments presented in this paper were carried out with the Moses toolkit [25], a state-of-the-art open-source phrase-based SMT system. The translation and the re-ordering model relied on “grow-diag-final” symmetrized word-to-word alignments

built using GIZA++ [26] and the training script of Moses. A 5-gram language model was trained using the IRSTLM toolkit [27], exploiting improved Modified Kneser-Ney smoothing, and quantizing both, probabilities and back-off weights.

4.3 Baseline

The baseline systems were trained with the toolkits and settings as described above. The in-domain baseline (IC-Baseline) was trained on the in-domain corpus which produces a 12.26M phrase table. The general-domain baseline (GC-Baseline) was substantially larger, having a 1.57G phrase table. The BLEU scores of the baseline systems are in Table 3.

The baseline results show that a translation system trained on the general corpus outperforms a system trained on the in-domain corpus by over 2.85 BLEU points. Although the in-domain data could improve the quality of word alignment, it is not broad enough to reduce the OOV words. As described in next section, the GC-Baseline system result will be further improved by data selection methods.

Table 3. BLEU via General and In-domain corpus

Baseline	Phrase Table Size	BLEU
<i>GC-Baseline</i>	1.57G	39.15
<i>IC-Baseline</i>	12.26M	36.30

5 Results and Discussion

In order to evaluate the performance of the presented models, we implemented three individual data selection models: cosine *tf-idf* (Cos-IR), bilingual cross-entropy difference (B-CED), fuzzy matching scorer (FMS) as well as the proposed model at corpus level (*iCPE-C*) and model level (*iCPE-M*). For each method, we selected the top $N=\{80K, 160K, 320K\}$ sentence pairs out of the 1.1M in the general corpus⁷. Table 4 contains BLEU scores of the systems trained on subsets selected via different models.

All the methods but FMS could be used to train a state-of-the-art SMT system. Cos-IR improves by at most 1.02 (offline) and 0.88 (online) BLEU points using 28.12% of the general corpus. The results approximately match with the conclusions of [9, 10]. This shows that keywords overlap plays a significant role in finding sentences in similar domains. Besides, Cos-IR has a strong robustness because the selection with online strategy still works well. However, it needs a large amount of selected data (28.0%) to obtain an ideal performance. The main reason is that the sentences including same keywords still may be irrelevant. For instance, there are two sentences including the same phrase “*according to the article*”, but one may be in the domain of law and other one may be from news.

⁷ Roughly 7.0%, 14.0%, 28.0% of general-domain corpus. Besides, *K* is short for thousand and *M* is short for million.

Table 4. Translation results via different methods

Method	Sentences	BLEU (Offline)	BLEU (Online)
<i>GC-Baseline</i>	1.1M	39.15	
<i>IC-Baseline</i>	1.1M	36.30	
<i>Cos-IR</i>	80K	39.04	37.53
	160K	39.85	39.45
	320K	40.17	40.03
<i>B-CED</i>	80K	40.91	35.50
	160K	41.12	39.47
	320K	40.02	40.98
<i>FMS</i>	80K	37.42	36.22
	160K	37.90	36.71
	320K	38.15	38.00
<i>iCPE-C</i>	80K	42.25	39.39
	160K	43.04	41.87
	320K	42.42	40.44
<i>iCPE-M</i>	80K	42.93	40.57
	160K	43.65	41.95
	320K	43.97	42.21

PPBased variant B-CED works very well with the offline strategy. It achieves 41.12 (using 7.0% data) and 40.98 (using 14.0% data) BLEU with offline and online strategies. This indicates that bilingual resources are very useful to build a stable in-domain model. When using an in-domain corpus as the reference set, B-CED should enlarge the size of selected data to obtain an ideal BLEU. It has a good but unstable performance with different strategies. The main reason is that considering the word order may be helpful to filter the noise, but it depends heavily upon the in-domain LMs. Similar to the discussion in Section 1, they are so sensitive to the quality and quantity of reference sets, which was not reported by [12].

FMS fails to outperform the baseline system even it is much stricter than other criteria. When adding word position factor into similarity measuring, FMS tries to find nearly the same sentences on length, collocation and semantics. But our general corpus seems not large enough to cover a certain amount of FMS-similar sentences. With increasing the size of general or in-domain corpus, we believe FMS may work better.

We combined Cos-IR, FMS and B-CED (which is the best one among PPBased criteria) and gave equal weights (set $\alpha = \beta = \lambda = 1$ in Eq. 4 and $\alpha_i = \beta_i = 1/3$ in Eq. 5 and 6) to each component at two combination levels. At both levels, iCPE performs much better than other methods as well as the baseline systems. This shows a strong ability to balance the OOV and noise problems. On the one hand, filtering too much unmatched words may not sufficiently address the data sparsity issue of the SMT model; on the other hand, adding too much of the selected data may lead to the dilution of the in-domain characteristics of the SMT model. However, it seems to succeed

the advantage of each individual model when combining them together. For instance, the performance of *iCPE* does not drop sharply (like PPBased approaches) when using an in-domain corpus as reference set. This not only shows its stronger robustness for building a real-life SMT system, but also proves that combination method works better than any single individual approach.

Furthermore, *iCPE* has achieved at most 3.89 (offline) and 2.72 (online) improvements over the baseline system at corpus level combination. Besides, the result is still higher than the best individual model (B-CED) by 1.92 (offline) and 0.91 (online). The performance can be further improved by interpolating at the model level. It works better (obtained around 1 BLEU point improvement) than the corpus combination method in the same settings.

6 Conclusions

In this paper, we regard data selection as a problem of measuring similarities via different criteria. This is the first time to systematically compare the state-of-the-art data selection methods for SMT adaptation. We not only explore edit-distance based method for this task for the first time, but also present offline and online strategies for fair comparison. We further integrate the presented individual data selection model at both corpus and model levels. It achieves a good performance in terms of its robustness and effectiveness.

In order to evaluate the proposed data selection model on a large general corpus, we compare it with 3 other related methods: Cos-IR, B-CED, FMS as well as two baseline systems. We can analyze the results from three different aspects:

- *Translation Quality.* The results show a significant performance of the most methods in particular the proposed *iCPE*. It suggests better to use bilingual resources in similarity measuring.
- *Noise Filtering.* *iCPE* could discard about 93% data of the general corpus with a better translation quality. While other models perform either badly or unsteadily.
- *Robustness.* To build a real-life system, in-domain data set is preferable (online strategy). However, only *iCPE* gives a consistently boosting performance.

Acknowledgment. The authors are grateful to the Science and Technology Development Fund of Macau and the Research Committee of the University of Macau for the funding support for our research, under the reference No. 017/2009/A and MYRG076(Y1-L2)-FST13-WF. The authors also wish to thank the anonymous reviewers for many helpful comments.

References

1. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2), 263–311 (1993)
2. Daumé III, H., Jagarlamudi, J.: Domain adaptation for machine translation by mining unseen words. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL HLT 2011 (2011)

3. Mansour, S., Ney, H.: A simple and effective weighted phrase extraction for machine translation adaptation. In: IWSLT (2012)
4. Koehn, P., Haddow, B.: Towards effective use of training data in statistical machine translation. In: Proceedings of the Seventh Workshop on Statistical Machine Translation, pp. 317–321 (2012)
5. Civera, J., Juan, A.: Domain adaptation in statistical machine translation with mixture modeling. In: Proceedings of the Second Workshop on Statistical Machine Translation, pp. 177–180 (2007)
6. Foster, G., Kuhn, R.: Mixture-model adaptation for SMT. In: Proceedings of the Second ACL Workshop on Statistical Machine Translation, pp. 128–136 (2007)
7. Eidelman, V., Boyd-Graber, J., Resnik, P.: Topic models for dynamic translation model adaptation. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, vol. 2, pp. 115–119 (2012)
8. Matsoukas, S., Rosti, A.V.I., Zhang, B.: Discriminative corpus weight estimation for machine translation. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 2, pp. 708–717 (2009)
9. Hildebrand, A.S., Eck, M., Vogel, S., Waibel, A.: Adaptation of the translation model for statistical machine translation based on information retrieval. In: Proceedings of EAMT, vol. 2005, pp. 133–142 (2005)
10. Lü, Y., Huang, J., Liu, Q.: Improving statistical machine translation performance by training data selection and optimization. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 343–350 (2007)
11. Moore, R.C., Lewis, W.: Intelligent selection of language model training data. In: Proceedings of the ACL 2010 Conference Short Papers, pp. 220–224 (2010)
12. Axelrod, A., He, X., Gao, J.: Domain adaptation via pseudo in-domain data selection. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 355–362 (2011)
13. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10, 707 (1966)
14. Koehn, P., Senellart, J.: Convergence of translation memory and statistical machine translation. In: Proceedings of AMTA Workshop on MT Research and the Translation Industry, pp. 21–31 (2010)
15. Leveling, J., et al.: Approximate sentence retrieval for scalable and efficient example-based machine translation. In: COLING 2012, pp. 1571–1586 (2012)
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318 (2002)
17. Wang, L.Y., Wong, D.F., Chao, L.S.: TQDL: Integrated models for cross-language document retrieval. *International Journal of Computational Linguistics and Chinese Language Processing (IJCLCLP)* 17(4), 15–31 (2012)
18. Wang, L.Y., Wong, D.F., Chao, L.S.: An improvement in cross-language document retrieval based on statistical models. In: Processing of the 24th Conference on Computational Linguistics and Speech (ROCLING 2012), pp. 144–155 (2012)
19. Stolcke, A., et al.: SRILM—an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing, vol. 2, pp. 901–904 (2002)
20. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, pp. 310–318 (1996)

21. Wang, L.Y., Wong, D.F., Chao, L.S., Xing, J.W., Lu, Y., Isabel, T.: Edit Distance: A new data selection criterion for SMT domain adaptation. In: *Proceedings of Recent Advances in Natural Language Processing* (2013)
22. Zhang, H.P., Yu, H.K., Xiong, D.Y., Liu, Q.: HHMM-based Chinese lexical analyzer ICTCLAS. In: *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, vol. 17, pp. 184–187 (2003)
23. Wang, L.Y., Wong, D.F., Chao, L.S., Xing, J.W.: CRFs-based Chinese word segmentation for micro-blog with small-scale data. In: *Proceedings of the Second CIPS-SIGHAN Joint Conference on Chinese Language*, pp. 51–57, December 20-21 (2012)
24. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: *MT Summit*, vol. 5 (2005)
25. Koehn, P., et al.: Moses: Open source toolkit for statistical machine translation. In: *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 177–180 (2007)
26. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 19–51 (2003)
27. Federico, M., Bertoldi, N., Cettolo, M.: IRSTLM: an open source toolkit for handling large scale language models. In: *Proceedings of Interspeech*, pp. 1618–1621 (2008)

Multi-classifier Combination for Translation Error Detection

Jinhua Du¹, Junbo Guo², Sha Wang¹, and Xiyuan Zhang¹

¹ Faculty of Automation and Information Engineering
jhd@xaut.edu.cn

² Faculty of Advanced Technology,
Xi'an University of Technology, Xi'an, China

Abstract. This paper proposes a multi-classifier combination strategy to improve translation error detection performance for statistical machine translation (SMT). Specifically, two different classifiers – Maximum Entropy (MaxEnt) and Support Vector Machine (SVM) – over different features perform a binary classification and export classification probabilities for either class. Then a probability product rule based multi-classifier combination strategy is employed to fuse these two classifiers to decrease the classification error rate (CER). Three typical word posterior probabilities (WPP) and three linguistic features as well as their combinations are used in the experiments conducted on Chinese-to-English NIST data sets. Experimental results show that the combination of multiple classifiers reduce the CER by relative 0.15%, 0.94%, and 1.52% compared to the SVM classifier, and relative 1.73%, 1.72%, 2.02% compared to the MaxEnt classifier over three different feature combinations.

Keywords: translation error detection, binary classification, multi-classifier combination.

1 Introduction

In recent years, a number of different types of SMT methods have been proposed, such as the phrase-based, hierarchical phrased-based, and syntax-based models etc., which significantly improve the translation quality. Meanwhile, a lot of effort has been put to apply SMT systems to practical use, e.g. the software localization industry [1–3]. However, the translation quality cannot fully satisfy the actual demand of industry yet. For example, the ungrammatical errors and disordered words in the translation often increase human cost. Therefore, high-quality automatic translation error detection or word-level confidence estimation is necessary to further improve the working efficiency of the post-editors or translators in the localization industry.

Typically, most translation error detection methods utilize system-based features (e.g. WPP) combining with extra knowledge such as linguistic features to decrease the classification error rate [4–9]. As to the system-based features,

a number of different algorithms to calculate the WPP were proposed based on the N -best list or word lattice, and had been applied to SMT translation quality estimation. Afterwards, some researchers try to introduce more useful knowledge sources such as syntactic and semantic features to further improve the error detection capability [8, 10, 11]. However, these features are not that easy to extract due to their complexity, low generalization capability, and dependency on specific languages etc. Hence, currently the system-based features such as WPP and lexicalized features (e.g. word and part-of-speech (POS)) still play the main role in the error detection task or the confidence estimation task.

Generally, translation error detection can be regarded as a binary classification task. Thus, the accuracy of the classifier also plays an important role in terms of improving the prediction capability besides adding new features and extra knowledge. This paper mainly focuses on the investigation of different classifiers, and presents an effective and straightforward strategy of combining two different classifiers to improve the classification performance. Firstly, we introduce the features used in our task, which are three typical WPP system-based features and three linguistic features, then employ two different classifiers, namely the MaxEnt classifier and SVM classifier to perform the classification task respectively. Finally, we carry out a combination operation – multiplication of the classification probabilities – to obtain the final result. Experiments are conducted on NIST Chinese-to-English translation task, and the results show that the combined method outperforms either individual classifier used in our task in terms of the CER.

The rest of the paper is organized as follows: Section 2 briefs the related work as to the error detection task. In Section 3, three typical WPP and three linguistic features are described. Section 4 firstly describes the MaxEnt and SVM classifiers used in our task, and then the multi-classifier strategy and feature representation are detailed. Experimental settings, implementation and analysis are reported in Section 5. The final section concludes and gives avenues for future work.

2 Related Work

The question of translation confidence estimation has attracted a number of researcher due to its importance in promoting SMT application. In 2004, Blatz et al. improved the basic confidence estimation method by combining the neural network and a naive Bayes classifier to predict the word-level and the sentence-level translation errors [6]. The features they used include WPP calculated from the N -best list, translation model-based features, semantic feature extracted from the WordNet, as well as simple syntactic features. Experimental results show that all among these features, WPP is more effective with strong generalization capability than linguistic features.

Ueffing and Ney exhaustively explore various kinds of WPP features to perform confidence measures, and proposed different WPP algorithms to verify the effectiveness in confidence estimation task [5, 7]. In their task, the words in the generated target sentence can be tagged as *correct* or *false* to facilitate post-editing or work in an interactive translation environment. Their experiments

conducted on different data sets show that different WPP algorithms perform differently, but basically each can reduce the CER. Furthermore, the combination of different features can perform better than any individual features.

Specia et al. have done a lot of work with regard to the confidence estimation in the computer-aided translation field [10, 11]. They categorize translations into “bad” or “good” classes based on sentence-level binary scores of the post-edition MT fragments. The features used are called “black-box” features, which can be extracted from any MT systems only if the information from the input (source) and translation (target) sentences are given, such as source and target sentence lengths and their ratios, the edit distance between the source sentence and sentences in the corpus used to train the SMT system. Their work contributed significantly to SMT translation confidence estimation research and application.

Xiong et al. proposed an MaxEnt classifier based error detection method to predict translation errors (each word is tagged as *correct* or *incorrect*) by integrating a WPP feature, a syntactic feature extracted from LG parser and some lexical features [8]. The experimental results show that linguistic features can reduce CER when used alone, and it outperforms WPP. Moreover, linguistic features can further provide complementary information when combined with WPP, which collectively reduce the classification error rate.

In 2011, Bach et al. classified translation errors into four categories by extracting more richer set of source-side information features, and combined sentence-level and word-level features to estimate translation quality. They predict error types of each word in the MT output with a confidence score, then extend it to the sentence level, and finally apply it to N -best list re-ranking task to improve MT quality [9].

On the basis of previous work, this paper mainly focuses on how to significantly improve the classification performance by using different kinds of classifiers and combining multiple classifiers over a set of effective features. Specifically, this paper 1) verifies the performance of various classifiers, namely the MaxEnt classifier and the SVM classifier on the translation error detection task; 2) presents a probability product combination strategy to fuse two classifier to obtain better results.

3 Features

3.1 WPP Feature

WPP is served as a major and effective confidence estimation feature both in speech recognition and in SMT post-processing. In terms of SMT, WPP refers to the probability of a word occurring in the hypothesis given a source input. Generally speaking, the underlying idea is that if the posterior probability of a word occurring in a hypothesis is high, then the chance that it is believed to be correct is big correspondingly. Based on this consideration, it is reasonable that the more useful information considered in the WPP algorithm, the better the performance would achieve.

The general mathematical description of WPP is as:

For an SMT system S , given the input sentence f_1^J , and the exported N -best list $e_{n,1}^{n,I_n}$, where $n = 1, \dots, N$, e_n refers to the n^{th} hypothesis with the probability $p(f_1^J, e_{n,1}^{n,I_n})$, then the WPP in the error detection task can be represented as calculating the probability $p_i(e|f_1^J, e_1^I)$ of the word e at position i in the 1-best hypothesis of the N -best list as in (1),

$$p_i(e|f_1^J, e_1^I) = \frac{\sum_{n=1}^N f(a, e_{n,i}, e) \cdot p(f_1^J, e_{n,1}^{n,I_n})}{\sum_{n=1}^N p(f_1^J, e_{n,1}^{n,I_n})} \quad (1)$$

where a is a hidden variable which indicates an alignment measure; $f(a, e_{n,i}, e)$ is a binary sign function as in (2),

$$f(a, e_{n,i}, e) = \begin{cases} 1 & e_{n,i} = e \\ 0 & otherwise \end{cases} \quad (2)$$

It can be seen from the description of N -best based WPP algorithm that the posterior probability of a word in a hypothesis can be worked out according to the sentence-level posterior probabilities of hypotheses in the N -best list. The vital information to be considered is the position of the word e which is determined by the alignment measure between the 1-best hypothesis and the rest of the N -best list.

Here we introduces three typical WPP methods to illustrate their different influence on the error detection performance over different kinds of classifiers.

3.1.1 Fixed Position Based WPP

The basic idea of fixed position-based WPP is that given an input f_1^J , the posterior probability of a word e at position i in the hypothesis e_1^I can be calculated by summing the posterior probabilities of all sentences in the N -best list containing target word e at target position i , which is as in (3),

$$p_i(e|f_1^J, e_1^I) = \frac{\sum_{n=1}^N \delta(e_{n,i}, e) \cdot p(f_1^J, e_{n,1}^{n,I_n})}{\sum_{e'} \sum_{n=1}^N \delta(e_{n,i}, e') \cdot p(f_1^J, e_{n,1}^{n,I_n})} \quad (3)$$

where $\delta(x, y)$ is the Kronecker function as in (4),

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & otherwise \end{cases} \quad (4)$$

This method only uses the original position information of each word without any extra alignment measure between the 1-best and any other hypotheses.

3.1.2 Flexible Position Based WPP

The potential problem of fixed position based WPP is that generally the hypotheses in the N -best list have different length that will make the same word occur at different positions so that the WPP would have a large error compared to the real probability distribution. Naturally the intuition to improve this method is to make the position flexible, e.g. using a sliding window.

The basic idea of sliding window is to consider the words around the position i , i.e., the context. Let the window size be t , then the sliding window at position i can be denoted as $i \pm t$. If the target word e appears inside the window, then we regard it occurring at position i and sum up the probability of the current hypothesis, which is formulated as in (5),

$$p_{i,t}(e|f_1^J, e_1^I) = \sum_{k=i-t}^{i+t} p_k(e|f_1^J, e_1^I) \quad (5)$$

where $p_k(e|f_1^J, e_1^I)$ is as illustrated in Eq. (3).

3.1.3 Word Alignment Based WPP

The sliding window based method needs to choose a proper window size which can only be determined by experiments. Thus, another straightforward way to improve the fixed position method is to perform the word alignment between the 1-best hypothesis and the rest of hypotheses in the N -best list, i.e., align the rest of hypotheses against the 1-best hypothesis.

Specifically, let $L(e_1^I, e_{n,1}^{n,I_n})$ be the Levenshtein alignment between e_1^I and other hypotheses, then the WPP of the word e at position i is as in (6):

$$p_{lev}(e|f_1^J, e_1^I) = \frac{p_{lev}(e, f_1^J, e_1^I)}{\sum_{e'} p_{lev}(e', f_1^J, e_1^I)} \quad (6)$$

where

$$p_{lev}(e, f_1^J, e_1^I) = \sum_{n=1}^N \delta(e, L_i(e_1^I, e_{n,1}^{n,I_n})) \cdot p(f_1^J, e_{n,1}^{n,I_n}) \quad (7)$$

In Eq. (7), $p(f_1^J, e_{n,1}^{n,I_n})$ is the posterior probability of each hypothesis in the N -best list, which is given by the SMT system. $\delta(x, y)$ is the Kronecker function as in Eq. (4).

3.2 Linguistic Features

3.2.1 Syntactic Features

Xiong et al. extracted syntactical feature by checking whether a word is connected with other words from the output of the LG parser. When the parser fails to parse the entire sentence, it ignores one word each time until it finds linkages for remaining words. After parsing, those ignored words which are not

connected to any other words to be called *null-linked* words. These *null-linked* words are prone to be syntactically incorrect and the linked words are prone to be syntactically correct, then a binary syntactic feature for a word according to its links can be defined as in (8),

$$\text{link}(e) = \begin{cases} \text{yes} & \mathbf{e} \text{ has links with other words} \\ \text{no} & \text{otherwise} \end{cases} \quad (8)$$

Refer to detailed description in [8].

3.3 Lexical Features

Lexical features such as the word itself and the POS [12] are common features used in NLP tasks. In this paper, we also utilize the word/pos with its context (e.g. the previous two words/pos and next two words/pos) to form a feature vector as follows,

- *word*: $(w_{-2}, w_{-1}, w, w_1, w_2)$
- *pos*: $(pos_{-2}, pos_{-1}, pos, pos_1, pos_2)$

4 Classifiers and Feature Representation

In this paper, the translation error detection is regarded as a classification task. In this section, we introduce two kinds of classifiers – MaxEnt and SVM, and then come up with a multi-classifier combination strategy to perform our translation error detection task.

Our translation error detection is a binary classification task that annotates a word e of the translation hypothesis e_1^l as “*correct*” if it is translated correctly, or “*incorrect*” if it is a wrong translation. Therefore, the label set for the classification task can be denoted as $\mathbf{y} = \{c, i\}$, where \mathbf{y} indicates the label set, c stands for class “*correct*” and i represents class “*incorrect*”.

4.1 Maximum Entropy Classifier

The MaxEnt model is the most commonly-used classifier in NLP tasks, which is a generalization of the model used by the naive Bayes classifier. The basic idea of MaxEnt model is to build a consistent model for all known factors without considering any unknown factors. A remarkable characteristic of the MaxEnt classifier is that features are not necessarily required independent. In doing so, features can be arbitrarily added into the model. Denote the binary classification samples as $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ (\mathbf{x}_i represents the feature vector, $y_i \in \{c, i\}$) (c and i stand for *correct* and *incorrect* labels respectively in our task), then as in literature [8], the MaxEnt classifier for a word e in a hypothesis can be formulated as in (9),

$$p(y|\mathbf{x}) = \frac{\exp(\sum_i (\lambda_i f_i(\mathbf{x}, y)))}{\sum_y (\exp(\sum_i (\lambda_i f_i(\mathbf{x}, y))))} \quad (9)$$

where f_i is a feature function, λ_i is the weight of f_i , y is the class label, and \mathbf{x} is the feature vector.

4.2 SVM Classifier

SVM has been widely and successfully used in many NLP tasks, such as word sense disambiguation, name entity recognition etc. The basic principle of SVM is to find an optimal hyperplane to make the distance maximum between two classes. The classification task can be defined as in (10),

$$g(x) = \text{sign}\left[\sum_1^n a_i y_i K(x_i, x) + b\right] \quad (10)$$

where $g(x)$ is the optimal classification hyperplane, $K(x_i, x)$ is the kernel function.

4.3 Multi-classifier Combination

Multiple classifiers combination method has been applied in many NLP tasks, such as word sense disambiguation etc. Most of these applications have shown a considerable improvement over the performance of individual classifiers. Therefore, it leads us to consider implementing such a multiple classifier combination strategy for the translation error detection task as well.

In general, different types of classifiers would reflect different characteristics in the classification results, so that using classifier combination techniques can potentially achieve a better classification accuracy based on the assumption that the errors made by each of the classifiers are not identical, and if the combination strategy is appropriate, then the outcome might correct some errors.

Several effective ways of classifier combination techniques have been studied, such as probability distribution based method, vote-based method, rank-based method, linear/weighted linear combination method etc. [13–15]. Regarding this task, considering that we only have two different classifiers, a straightforward strategy – probability product rule – is presented to combine the outputs of two individual classifiers to achieve better results.

The algorithm of the probability product for our translation error detection task can be formalized as:

Assume the task is a binary classification, given the classifier set $C = \{C_1, \dots, C_n\}$ and the class set $c = \{c_1, c_2\}$, for a word sequence $S = \{w_1, \dots, w_m\}$ in which each word w_i need to be tagged as c_1 or c_2 , if the outputs for a word w from each individual classifier C_i can be denoted as $O_w^i = \{p_{c_1}^i, p_{c_2}^i\}$, in which $p_{c_1}^i$ indicates the probability that the word w is tagged as c_1 by the classifier C_i , and $p_{c_2}^i$ is the probability that w is tagged as c_2 by the classifier C_i , conditioned by $p_{c_1}^i + p_{c_2}^i = 1$, then the probability product algorithm for the classifier combination can be formulated as in (11),

$$c_w = \max\left\{\prod_{i=1}^n p_{c_1}^i, \prod_{i=1}^n p_{c_2}^i\right\} \quad (11)$$

where c_w indicates the predicted class for the word w . In our task, $n = 2$.

4.4 Feature Vector Representation

As described in previous sections, in our translation error detection task, we have four kinds of features: *wpp*, *pos*, *word* and *link* (c.f. Section 3). In this section, we introduce how to construct a normalized and unified feature vector format for the MaxEnt and SVM classifiers.

Generally in the NLP classification task, context information is usually to be considered in the process of feature extraction. Therefore, in our task, to build a feature vector for a word e , we look at 2 words before and 2 words after the current word position as well. Thus, the feature vector \mathbf{x} that includes four kinds of features can be denoted as,

$$\mathbf{x} = \langle wpp_{-2}, wpp_{-1}, wpp, wpp_1, wpp_2, pos_{-2}, pos_{-1}, pos, pos_1, pos_2, word_{-2}, word_{-1}, word, word_1, word_2, link_{-2}, link_{-1}, link, link_1, link_2 \rangle$$

As to the individual classifiers, we use the MaxEnt toolkit ¹ as our MaxEnt classifier, and use LibSVM ² as our SVM classifier [16] respectively.

5 Experiments and Analysis

5.1 Chinese-English SMT System

We utilize Moses [17] to provide 10,000-best list with translation direction from Chinese to English. The training data consists of 3,397,538 pairs of sentences (including Hong Kong news, FBIS, ISI Chinese-English Network Data and Xinhua news etc.). The language model is five-gram built on the English part of the bilingual corpus and Xinhua part of the English Gigaword.

The development set for SMT training is the current set of NIST MT 2006 (1,664 source sentences) and the test sets are NIST MT-05 (1,082 sentences) and NIST MT-08 (1,357 sentences). Each source sentence has four references. During the decoding process, the SMT system exports 10,000-best hypotheses for each source sentence, i.e., $N = 10,000$.

Performance of SMT systems on two test sets is shown in Table 1 in terms of BLEU4 and other metrics.

Table 1. SMT performance and the ratio of correct words (RCW)

dataset	BLEU4(%)	WER(%)	TER(%)	RCW(%)
NIST MT 2008	25.97	69.79	63.56	37.99
NIST MT 2005	33.17	69.50	61.40	41.59

¹ <http://homepages.inf.ed.ac.uk/s0450736/maxenttoolkit.html>.

² Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

5.2 Experimental Settings for Translation Error Detection Task

Development and Test Sets: in the error detection task, we use NIST MT-08 as the development set to tune the classifiers, and NIST MT-05 as the test set to evaluate the classification performance.

Data Annotation: we use the WER metric in TER toolkit [18] to determine the true labels for the words in the development set and the test set. Specifically, we firstly perform the minimum edit distance alignment between the hypothesis and the four references, and then select the one with minimum WER score as the final reference to tag the hypothesis. That is, a word e in the hypothesis is tagged as c if it is the same as that in the reference, otherwise tag it as i .

There are 14,658 correct words and 23,929 incorrect words in the 1-best hypothesis of MT-08 set (37.99% ratio of correct words, RCW), 15,179 correct words and 21,318 incorrect words in the 1-best hypothesis of MT-05 set (41.59% RCW). See RCW in Table 1.

Evaluation Metrics: the commonly-used evaluation metrics for the classification task includes CER (classification error rate), precision, recall and F measure. In our translation error detection task, we use CER as the main evaluation metric to evaluate the system performance that is defined as in (12),

$$\text{CER} = \frac{\text{\#of wrongly tagged words}}{\text{\#of total words}} \quad (12)$$

Since the RCW is less than 50% (41.59%), i.e., the number of incorrect words is more than correct words, it is reasonable to use the RCW as the baseline of CER to examine the classification performance of classifiers.

We also use F measure, Precision and Recall as the auxiliary evaluation metrics to evaluate some performance of features and classifiers. See definitions in [8].

5.3 Classification Experiments for Individual Classifiers

Results of different features and feature combinations over two different individual classifiers are shown in Table 2.

WPP_Dir represents the fixed position-based WPP, WPP_Win represents the flexible position-based WPP with the window size 2, and WPP_Lev represents word alignment-based WPP. $com1$ represents the feature combination of $WPP_Dir + Word + Pos + Link$, $com2$ stands for the feature combination of $WPP_Win + Word + Pos + Link$, and $com3$ indicates the feature combination of $WPP_Lev + Word + Pos + Link$.

We can see that 1) all these individual features over two classifiers significantly reduce the CER compared to the baseline; 2) the WPP_Win and WPP_Lev perform better than WPP_Dir which shows that position information is helpful; 3) linguistic features perform better than three WPP features over the MaxEnt (except $link$), while worse than those over the SVM classifier in terms of CER. However, they all significantly reduce the CER compared to the baseline; 4) $WPP_Win + word + pos + link$ obtains the best performance both on MaxEnt

Table 2. Results of two individual classifiers for translation error detection

Feature	MaxEnt				SVM			
	CER(%)	P(%)	R(%)	F(%)	CER(%)	P(%)	R(%)	F(%)
Baseline	<i>41.59</i>	–	–	–	<i>41.59</i>	–	–	–
WPP_Dir	<i>40.48</i>	63.44	72.46	67.65	37.64	61.19	97.20	75.11
WPP_Win	<i>39.70</i>	63.82	73.95	68.51	37.47	61.31	97.17	75.18
WPP_Lev	<i>40.12</i>	60.24	92.07	72.83	37.37	61.37	97.24	75.25
word	<i>39.11</i>	64.20	76.67	69.04	37.68	64.06	80.84	71.48
pos	<i>39.50</i>	61.52	86.46	71.89	39.12	62.10	84.75	73.68
link	<i>40.89</i>	59.55	93.55	72.77	37.70	61.36	95.71	74.78
com1	<i>35.93</i>	63.93	88.30	74.17	35.36	63.93	90.57	74.95
com2	<i>35.55</i>	64.77	85.83	73.83	35.27	64.11	89.98	74.86
com3	<i>35.62</i>	65.31	83.22	73.15	35.44	64.02	89.81	74.75

and SVM classifiers. Feature combinations outperform any of the individual features; 5) SVM classifier outperforms the MaxEnt classifier on all features in terms of the CER.

5.4 Classification Experiment on Multi-classifier Combination Strategy

The results of the Multi-classifier Combination experiment are shown in Table 3.

Table 3. Results of multi-classifier combination for translation error detection

Feature	MaxEnt		SVM		Multi-classifier	
	CER(%)	F(%)	CER(%)	F(%)	CER(%)	F(%)
Baseline	<i>41.59</i>	–	<i>41.59</i>	–	<i>41.59</i>	–
com1	<i>35.93</i>	74.17	<i>35.36</i>	74.95	35.31	74.55
com2	<i>35.55</i>	73.83	<i>35.27</i>	74.86	34.94	74.55
com3	<i>35.62</i>	73.15	<i>35.44</i>	74.75	34.90	74.47

We can see from the results that 1) compared to the baseline, the proposed multi-classifier combination method achieved significant improvement by relative 15.10%, 15.99% and 16.09% in terms of CER. 2) compared to the MaxEnt and SVM classifiers over three feature combinations, namely *com1*, *com2* and *com3*, the proposed multi-classifier combination method achieved significant improvement respectively by relative 0.15%, 0.94%, 1.52%, and 1.73%, 1.72%, 2.02% in terms of CER. 3) *WPP_Lev + word + pos + link* and *WPP_Win + word + pos + link* are significantly better than *WPP_Dir + word + pos + link*, which indicates that the flexible position based WPP feature is more useful than the fixed position based WPP on the multi-classifier combination.

From the comparison of the results, we can conclude: 1) generally speaking, *WPP_Win* performs the best and robust both in the three individual WPP

features and the three combined features. The reason we consider is that the sliding window makes the alignment more flexible and considers more context information. 2) linguistic features are helpful to the error detection. 3) multi-classifier strategy is effective to further improve the error detection performance.

Based on the observations, we also found that 1) the name entities (person name, location name, organization name etc.) are prone to be wrongly classified; 2) the prepositions, conjunctions, auxiliary verbs and articles are easier to be wrongly classified due to the factors that they often have an impact on the word orders or lead to empty alignment links; 3) the proportion of the notional words that are wrongly classified is relatively small.

Conclusions and Future Work

This paper presents a multi-classifier combination strategy for translation error detection. Firstly three different kinds of WPP features, three linguistic features and two individual classifiers are introduced, then a probability product based multi-classifier combination method is proposed which multiplies the corresponding classification probabilities respectively coming from MaxEnt and SVM classifiers for each word in a hypothesis, and then decides the label by the maximum probability. Experimental results on Chinese-to-English NIST MT data sets show that 1) in terms of individual classifiers used in our experiments, SVM classifier outperforms the MaxEnt classifier; 2) the proposed multi-classifier combination method performs the best compared to two individual classifiers.

In future work, we intend to carry out further study on the error detection task in the respects of 1) introducing paraphrases to annotate the hypotheses so that it can truly reflect the *correct* or *incorrect* at the semantic level; 2) introducing new useful features to further improve the detection capability; 3) performing experiments on more language pairs to verify our proposed method.

Acknowledgments. This work is supported by NSF project (61100085), SRF for ROCS, State Education Ministry, and Research Foundation of Education Department of Shaanxi Provincial Government (11JK1029). Thanks the reviewers for their insightful comments and suggestions.

References

- [1] DeCamp, J.: What is missing in user-centric MT? In: Proceedings of MT Summit XII, pp. 489–495 (2009)
- [2] Roturier, J.: Deploying novel MT technology to raise the bar for quality: a review of key advantages and challenges. In: Proceedings of MT Summit XII, pp. 1–8 (2009)
- [3] Simard, M., Isabelle, P.: Phrase-based machine translation in a computer-assisted translation environment. In: Proceedings of MT Summit XII, pp. 120–127 (2009)
- [4] Gandrabur, S., Foster, G.: Confidence Estimation for Translation Prediction. In: Proceedings of the HLT-NAACL, pp. 95–102 (2003)

- [5] Ueffing, N., Macherey, K., Ney, H.: Confidence Measures for Statistical Machine Translation. In: Proceedings of MT Summit IX, pp. 169–176 (2003)
- [6] Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kuesza, A., San-chis, A., Ueffing, N.: Confidence Estimation for Machine Translation. In: Proceedings of COLING, pp. 315–321 (2004)
- [7] Ueffing, N., Ney, H.: Word-Level Confidence Estimation for Machine Translation. *Computational Linguistics* 33(1), 9–40 (2007)
- [8] Xiong, D., Zhang, M., Li, H.: Error detection for statistical machine translation using linguistic features. In: Proceedings of ACL, pp. 604–611 (2010)
- [9] Bach, N., Huang, F., Al-Onaizan, Y.: Goodness: A Method for Measuring Machine Translation Confidence. In: Proceedings of ACL, pp. 211–219 (2011)
- [10] Specia, L., Cancedda, N., Dymetman, M., Turchi, M., Cristianini, N.: Estimating the sentence-level quality of machine translation systems. In: Proceedings of EAMT, pp. 28–35 (2009)
- [11] Specia, L., Saunders, C., Turchi, M., Wang, Z., Shawe-Taylor, J.: Improving the confidence of machine translation quality estimates. In: Proceedings of MT Summit, pp. 136–143 (2009)
- [12] Ratnaparkhi, A.: A Maximum Entropy Model for Part-of-Speech Tagging. In: Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP), Philadelphia, pp. 133–142 (1996)
- [13] Battiti, R., Colla, A.: Democracy in Neural Nets: Voting Schemes for Classification. *Neural Networks* 7(4), 691–707 (1994)
- [14] Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 20(3), 226–239 (1998)
- [15] Florian, R., Cucerzan, S., Schafer, C., Yarowsky, D.: Combining Classifiers for word sense disambiguation. *Natural Language Engineering* 8(4), 327–341 (2002)
- [16] Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 3(2), 27:1–27:27 (2011)
- [17] 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: Proceedings of ACL, pp. 177–180 (2010)
- [18] Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J.: A study of translation edit rate with targeted human annotation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, MD, pp. 223–231 (2006)

Automatic Discrimination of Pronunciations of Chinese Retroflex and Dental Affricates

Akemi Hoshino¹ and Akio Yasuda²

¹ Toyama National College of Technology, Ebie, Neriya, Imizu-city, Toyama, Japan
hoshino@nc-toyama.ac.jp

² Tokyo University of Marine Science and Technology, Etchujima, Koko-ku, Tokyo, Japan
yasuda@kaiyodai.ac.jp

Abstract. Retroflex aspirates in Chinese are generally difficult for Japanese students learning pronunciation. In particular, discriminating between utterances of aspirated dental and retroflex syllables is the most difficult to learn. We extracted the features of correctly pronouncing the aspirated dental syllables ca[ts'a], ci[ts'i], and ce[ts'ɤ] and aspirated retroflex ones cha[ts'a], chi[ts'i], and che[ts'ɤ] by observing the spectrum evolution of breathing power during both voice onset time and voiced period of sounds uttered by nine Chinese native speakers. We developed a 35-channel filter bank on a personal computer to analyze the evolution of breathing power spectrum by using MATLAB. We then automatically evaluated the utterances of 20 students judged to be correct by native Chinese speakers and obtained a success rate of higher than 90% and 95% for aspirated retroflex and dental syllables, respectively.

Keywords: Chinese aspirated retroflex and dental syllables, pronunciation training.

1 Introduction

Retroflex aspirates in Chinese are generally difficult for Japanese students learning pronunciation, because the Japanese language has no such sounds. In particular, discriminating between utterances with aspirated dental and retroflex syllables is the most difficult to learn. We observed a classroom of Japanese students of Chinese uttering aspirated retroflex sounds modeled after examples uttered by a native Chinese instructor. However, the utterances sounded like dental syllables to the instructor, and many students could not produce the correct sounds. They could not curl their tongues enough to articulate correctly, because there is no retroflex sounds in Japanese syllables.

We previously [1,2,3,4,5] showed that the breathing power during voice onset time (VOT) is a useful measure for evaluating the correct pronunciation of Chinese aspirates. We also developed an automatic evaluation system [6,7] for the students pronouncing Chinese aspirated syllables in accordance with the two parameters of VOT length and the breathing power during VOT.

However, since the system does not quite discriminate between aspirated retroflex and the dental syllables, we extracted the features of correctly pronouncing the aspirated dental syllables $ca[t\text{ʂ}'a]$, $ci[t\text{ʂ}'i]$, and $ce[t\text{ʂ}'\text{ɤ}]$ and the aspirated retroflex ones $cha[t\text{ʂ}'a]$, $chi[t\text{ʂ}'i]$, and $che[t\text{ʂ}'\text{ɤ}]$ by analyzing the spectrum of breathed power during VOT of sounds uttered by Chinese native speakers. For this research, we developed a 35-channel frequency filter bank by using a personal computer. We found that the main difference between aspirated dental and retroflex syllables appeared in the spectrogram of the breathed power during VOT [8].

To improve the discrimination of these syllables, we extracted the features of correctly pronouncing aspirated dental affricate and aspirated retroflex syllables by analyzing the frequency spectrum of breathed power during both VOT and inside the voiced period of sounds and established improved evaluation criteria. We discuss the results of successfully discriminating between aspirated dental affricate and aspirated retroflex syllables by Japanese students.

We will continue to apply our system to other Chinese aspirated syllables to develop automatic training system.

2 Difference between Aspirated Dental Affricate and Aspirated Retroflex Syllables

The affricate is a complex sound generated by simultaneously articulating explosive and fricative sounds as one sound in the same point of articulation.

The dental sound $[t\text{ʂ}']$ of the aspirate of Chinese is called the alveolar affricate and is formed by articulating an explosive and fricative sound at the point between the tip of the tongue and teeth.

The Chinese aspirated retroflex sound $[t\text{ʂ}']$ is also called the sublamino-postalveolar affricate, and the point of articulation is the sublamino-postalveolar. In articulating it, one curls the tongue firmly and articulates explosive and fricative sounds simultaneously.

In this chapter, we define the distinctive features that discriminate between the dental affricate $[t\text{ʂ}']$ and retroflex one $[t\text{ʂ}']$ by examining the spectrogram of the pairs $ca[t\text{ʂ}'a]$ - $cha[t\text{ʂ}'a]$, $ci[t\text{ʂ}'i]$ - $chi[t\text{ʂ}'i]$, and $ce[t\text{ʂ}'\text{ɤ}]$ - $che[t\text{ʂ}'\text{ɤ}]$ uttered by a native Chinese speaker.

Figure 1 shows the temporal evolution of spectrograms of the aspirated retroflex sound $cha[t\text{ʂ}'a]$ (left) and the aspirated dental sound $ca[t\text{ʂ}'a]$ (right) uttered by a Chinese speaker. The lower part of the figure shows the waveform of the voltage evolution picked up by a microphone. The ordinate extended upward shows the frequency component and the shade of the stripes implies the approximate power level at the corresponding time and frequency. The aspirate appears in the brief interval in the right spectrogram of $ca[t\text{ʂ}'a]$, indicated by light and thin vertical stripes during VOT, between the stop burst and the onset of vocal fold vibrations followed by a vowel

sound. This time interval is called the VOT [9], which is long, 160 ms. Although slightly darker stripes appear between 2500 and 5000 Hz in frequency and 70 and 150 ms in VOT, the temporal variation in the breathing power during VOT is not significant.

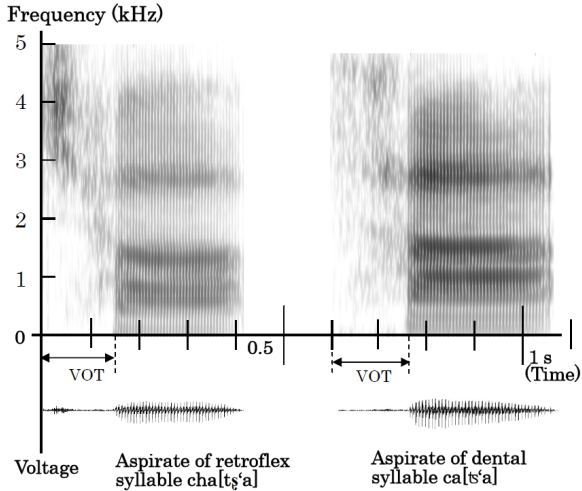


Fig. 1. Spectrograms of retroflex aspirate syllable cha[tʂ'a] (left) and dental aspirated syllable ca[tʂ'a] (right) pronounced by Chinese speaker

The left spectrogram is for the aspirated retroflex sound cha[tʂ'a] uttered by a Chinese speaker. The VOT was long, 150 ms. The dark vertical stripes in the upper left were observed between 2500 and 5000 Hz in frequency, during 0~70 ms of VOT. This is caused by friction of breath during breath release, which arises at a spot between the curled tongue and posterior alveolar. The large energy in the mouth dissipates at the early stage of VOT and generates high breathing power there. The thick horizontal bands in the voiced period in the right part of the spectrogram imply the formants that help to discriminate between the three dental syllables. The criteria are discussed later.

Figure 2 shows the temporal variation in spectrograms of the aspirated retroflex sound chi[tʂ'i] (left) and the aspirated dental sound ci[tʂ'i] (right) uttered by a Chinese speaker. The VOT of the aspirated dental sound ci[tʂ'i] was long, 225 ms, on the right hand side of the spectrogram. The unvarying darkness of the vertical bands shows that breathing power was rather steady during VOT. The left spectrogram is for the aspirated retroflex sound chi[tʂ'i]. The VOT was long, 250 ms.

During almost the entire VOT, the dark vertical stripes were observed in the frequencies between 2000~5000 Hz. This is due to the friction of breath at the breath release, which arises at a spot between the curled tongue and posterior alveolar.

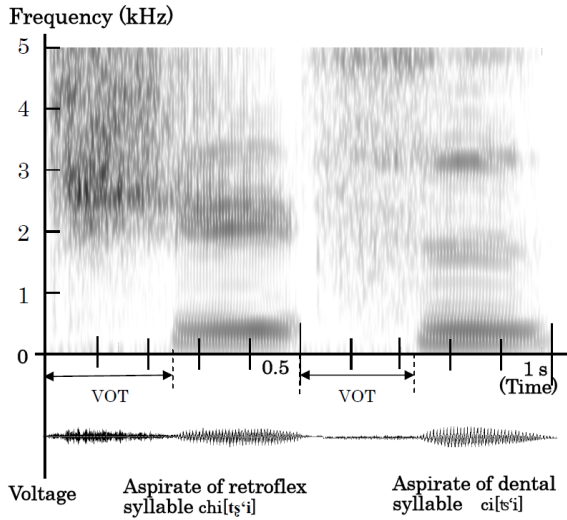


Fig. 2. Spectrograms of retroflex aspirate syllable $chi[tʂ'i]$ (left) and dental aspirated syllable $ci[tʂ'i]$ (right) pronounced by Chinese speaker

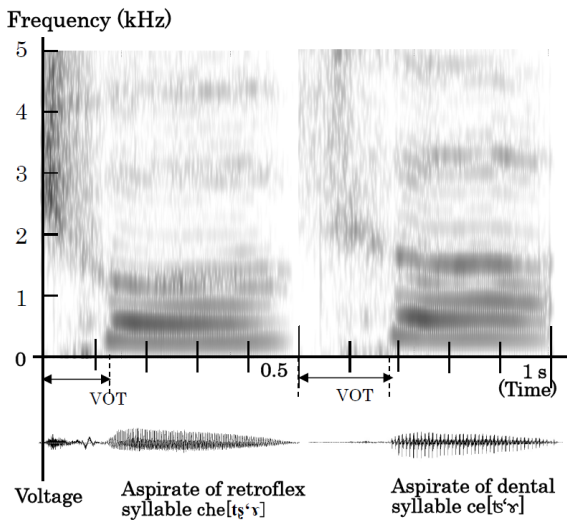


Fig. 3. Spectrograms of retroflex aspirate syllable $che[tʂ'ɤ]$ (left) and dental aspirated syllable $ce[tʂ'ɤ]$ (right) pronounced by Chinese speaker

Figure 3 shows the temporal variation in spectrograms of the aspirated retroflex sound $che[tʂ'ɤ]$ (left) and the aspirated dental sound $ce[tʂ'ɤ]$ (right) uttered by a Chinese speaker. The VOT of the aspirated dental sound $ce[tʂ'ɤ]$ was long, 180 ms. The stripes above 2000 Hz are darker and imply slightly stronger breathing power there.

For the frequency lower than 1200 Hz in VOT, the vertical stripes are light in accordance with weak breathing power. The distinctive feature of retroflex aspirated syllables is that they have a non-uniform spectrum in frequency and/or time during VOT, whereas aspirated dental ones have a rather uniform spectrum, as shown in the right spectrogram.

3 Automatic Measurement of VOT and Breathing Power

We showed that the correct utterance of retroflex aspirate and dental aspirated syllables is closely related to the frequency spectrum in VOT.

We previously developed an automatic measurement system of VOT and the breathing power by using a personal computer containing a 35-channel frequency filter bank, designed using MATLAB, in which the center frequency ranged from 50 to 6850 Hz with a bandwidth of 200 Hz [6,7]. We can extract the features of aspirated retroflex affricate and aspirated dental affricate syllables of the frequency spectrum in both VOT and voiced periods.

3.1 VOT Measurement Algorithm

We automatically detected the onset of burst. Pronounced signals were introduced into the filter bank and split into the power at each center frequency every 5 ms. The start time of VOT, t_1 , was determined by comparing the powers for the adjacent time frames when the number of temporally increasing channels was maximum. The end of VOT, t_2 , was the start point of the formant. Thus, t_2-t_1 is defined as VOT.

We described the features of correct pronunciation of aspirated dental and retroflex affricate syllables by observing the temporal variation of breathing power spectrum during VOT in Chapter 2. The powers at each frequency of the 35 channels every 5 ms with 11.025kHz sampling were added in accordance with the frequency criteria defined in Chapter 2 during VOT.

3.2 Breathing Power Measurement Algorithm

The average power during VOT is defined as follows. The powers are deduced every 5ms and are referred to as $P_{i,j}$, which is the power at $j \times 5\text{ms}$ of the $i(1-35)$ -channel where P_i is the integration of the power at each time in VOT of the i -channel, as shown in Equation (1).

$$P_i = \sum_{j=1}^J P_{i,j}(t_j) \quad (1)$$

Thus the energy W_i of the i -channel is defined as

$$W_{i,VOT} = P_i \times 5\text{ms} \quad (2)$$

The average power, $P_{i,av}$, of each frequency channel during VOT is defined as

$$P_{i,av} = W_{i,VOT} / VOT \quad (3)$$

The average power at i-channel in voiced period, T_{vs} , $P_{vi,av}$ can be defined similarly as

$$P_{vi,av} = W_{i,vs} / T_{vs} \quad (4)$$

4 Relationship between Breathing Power and Its Frequency Dependency during VOT and Quality of Pronunciations

Although several reports [9,10] on voiced retroflex have been published, there have been few reports on aspirated retroflex. We define the discrimination criteria of aspirated dental affricate and aspirated retroflex syllables by examining the VOT and the breathing power spectrum during VOT of pronunciation of the pairs $ca[t\zeta'a]$ - $cha[t\zeta'a]$, $ci[t\zeta'i]$ - $chi[t\zeta'i]$, and $ce[t\zeta'ɤ]$ - $che[t\zeta'ɤ]$ uttered by 20 Japanese students. We used our automatic measuring system to define the parameters.

4.1 Scoring of Pronunciation Quality of Students

To investigate the correct pronunciation criteria of the aspirated retroflex affricate syllables $cha[t\zeta'a]$, $chi[t\zeta'i]$, and $che[t\zeta'ɤ]$ and the aspirated dental ones $ca[t\zeta'a]$, $ci[t\zeta'i]$, and $ce[t\zeta'ɤ]$, the sounds uttered by 20 Japanese students were ranked using a listening test of the reproduced sounds conducted by nine native Chinese speakers [1-7]. The scores were as follows: 3 = correctly pronounced aspirated retroflex affricate or aspirated dental syllable; 2 = unclear sounds; and 1 = pronunciation in which the aspirated retroflex sounds were judged to be aspirated dental sounds and vice versa. We defined an average score of more than 2.6 as good. This score corresponds to the case in which six examiners give a score of '3' and three give a score of '2'. The examiners checked with each other that their pronunciations were perfectly aspirated. Some data were excluded in cases of split evaluations and a standard deviation of larger than 0.64, broken sounds uttered very close to the microphone, and sounds with a low S/N uttered away from the microphone.

4.2 Relationship between Scoring of Student Pronunciation and Evaluation Parameters

We now discuss the distribution of the student data with their scores are displayed on the surface of VOT and power respectively on abscissa and ordinate.

Figure 4 shows the data distributions on the surface of VOT and power with the scores of student pronunciations of aspirated retroflex $cha[t\zeta'a]$ and aspirated dental $ca[t\zeta'a]$. The power of each utterance in this figure was automatically calculated at the frequencies between 2750 Hz (Channel-15) and 5750 Hz (Channel-29) averaged during the start time of VOT to $1/2VOT$. The pronunciations of $cha[t\zeta'a]$ with a good score gathered in the upper

right from the center of the figure. The uttering power of aspirated retroflex syllable cha[tʂ'a] increased by a continuous sequence of fricative articulations, and utterances with the power higher than 17 received scores higher than 2.6. In contrast, the utterances with insufficient curling of the tongue received a low score. The data with the power weaker than 16 were received a low score. As for the utterance of aspirated dental ca[tʂ'a], the data gathered downward a little from the middle of the figure. The data with the power of 8~12 scored higher than 2.8. The two data points of aspirated dental syllable ca[tʂ'a], located at the top left, received a low score, presumably because unnecessary curling of the tongue resulted in high power utterance.

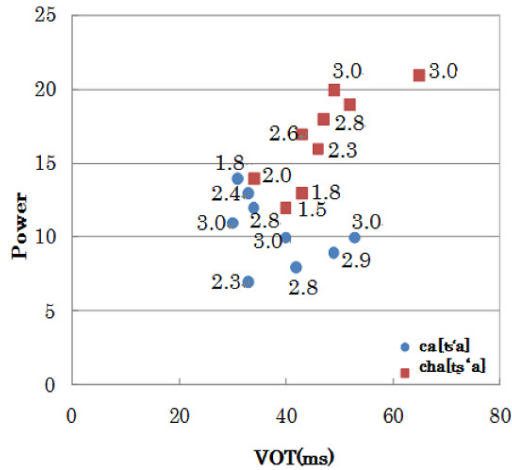


Fig. 4. Data distribution and scores for retroflex aspirated syllable cha[tʂ'a], and dental aspirated syllable ca[tʂ'a] with VOT on the abscissa and P_{av} at (2750-5750 Hz) on the ordinate

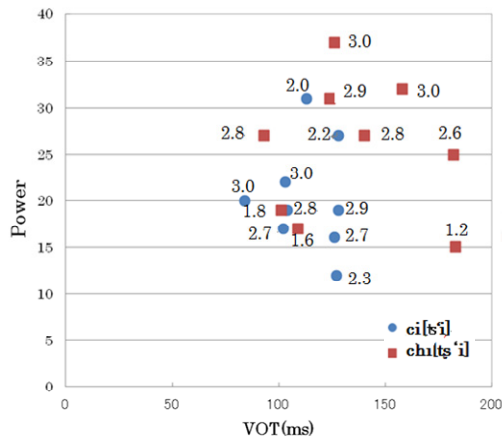


Fig. 5. Data distribution and scores for retroflex aspirated syllable cha[tʂ'i] and dental aspirated syllable ca[tʂ'i] with VOT on the abscissa and P_{av} at the frequencies between 1750 and 6350Hz on the ordinate

Figure 5 shows the data distributions on the surface of VOT and power with the scores of the student pronunciations of aspirated retroflex syllable $chi[t\zeta'i]$ and aspirated dental syllable $ci[t\zeta'i]$. The power of each utterance in this figure is summed one between the frequencies of 1750 Hz (Channel-10) and 6350 Hz (Channel-32) in VOT. The utterance with power higher than 25 of aspirated retroflex syllable $chi[t\zeta'i]$ receives a good score. Three utterance data points in the lower part of the figure, of aspirated retroflex syllable $chi[t\zeta'i]$ had utterance powers that are too low to pass the scoring test, i.e. 1.8, 1.6, and 1.2. As for utterances of aspirated dental syllable $ci[t\zeta'i]$, the data with powers of 16~22 obtained higher scores.

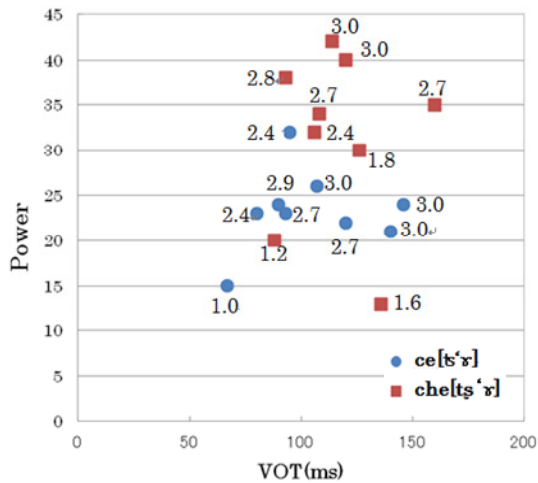


Fig. 6. Data distribution and scores for retroflex aspirated syllable $che[t\zeta'ɤ]$ and dental aspirated syllable $ce[t\zeta'ɤ]$ with VOT on the abscissa and P_{av} at the frequencies between 1150 and 5950Hz on the ordinate

Figure 6 shows the data distributions on the surface of VOT and power with the scores of the student pronunciations of aspirated retroflex syllable $che[t\zeta'ɤ]$ and aspirated dental syllable $ce[t\zeta'ɤ]$. The power of each utterance in this figure is summed one between the frequencies of 1150 Hz (Channel-7) and 5950 Hz (Channel-30) in VOT. Pronunciations of the aspirated retroflex syllable $chi[t\zeta'ɤ]$ with the power higher than 34 scored higher than 2.7. Pronunciations with the power lower than 32 were not correct. For the pronunciations of the aspirated dental syllable $ce[t\zeta'ɤ]$, the data with the power between 20~26 obtain successful scores.

5 Automatic Discrimination of Aspirated Retroflex Syllables and Dental Syllables

5.1 Parameters for Discrimination

Table 1 lists the evaluation criteria on utterances of retroflex aspirated syllables. If the power was higher than 17 between 2750 Hz (CH15) and 5750 Hz (CH29) averaged

during the onset of VOT to 1/2 of VOT, the utterances were judged to be aspirated retroflex syllable cha[t_ʂ'a]. If the power was higher than 25 between 1750 Hz (CH10) and 6350 Hz (CH32) throughout VOT, the utterances were judged to be aspirated retroflex syllable chi[t_ʂ'i]. If power was higher than 34 at the frequencies between 1150 Hz (CH7) and 5950 Hz (CH30) averaged during the onset of VOT to 2/3 of VOT, the utterances were judged to be aspirated retroflex syllable che[t_ʂ'ɤ].

Table 1. Evaluation criteria on utterance of retroflex aspirated syllables

Syllable	Channels(CH)	Frequency domain(Hz)	VOT range	Ave.Power in VOT
cha[t _ʂ 'a]	CH15~CH29	2750~5750	0~VOT/2	17 or more
chi[t _ʂ 'i]	CH10~CH32	1750~6350	Whole VOT	25 or more
che[t _ʂ 'ɤ]	CH07~CH30	1150~5950	0~VOT*2/3	34 or more

Table 2. Evaluation criteria on utterance of dental aspirated syllables of formant frequencies

Syllable	F1(Hz)/(CH)	F2(Hz)/(CH)	F3(Hz)/(CH)
ca[t _s 'a]	750~950/(CH5)	1150~1350/(CH7)	2150~2350/(CH12)
ci[t _s 'i]	150~350/(CH2)	1350~1550/(CH8)	2550~2750/(CH14)
ce[t _s 'ɤ]	350~550/(CH3)	1150~1350/(CH7)	2350~2550/(CH13)

Table 2 lists the evaluation criteria on the utterances of dental aspirated syllables, which depend on the formant frequency values of F1, F2, and F3. If high power appears between 750 and 950Hz, 1150 and 1350 Hz, and 2150 and 2350 Hz, the utterances were judged to be aspirated dental syllable ca[t_s'a]. If high power appeared between 150 and 350 Hz, 1350 and 1550 Hz, and 2550 and 2750 Hz, the utterances were judged to be aspirated dental syllable ci[t_s'i]. If high power appeared at the frequency channels between 350 and 550 Hz, 1150 and 1350 Hz, and 2350 and 2550 Hz, the utterances were judged to be aspirated dental syllable ce[t_s'ɤ].

5.2 Experiment and Results

We tried to discriminate between the pronunciations of the pairs ca[t_s'a] - cha[t_ʂ'a], ci[t_s'i] - chi[t_ʂ'i], and ce[t_s'ɤ] - che[t_ʂ'ɤ] uttered by 20 Japanese students. All utterances were evaluated to be correct by a listening test involving four native Chinese speakers.

Figure 7 illustrates the flow of our system for automatically discriminating aspirated retroflex and aspirated dental syllables. In step 1, the uttered sounds are input to the computer. In step 2, the sounds are automatically analyzed using our developed 32-channel filter bank to create a data base of the temporal variation of power spectrum. In step 3, VOT is deduced using the algorithm described in Subsection 3.1.

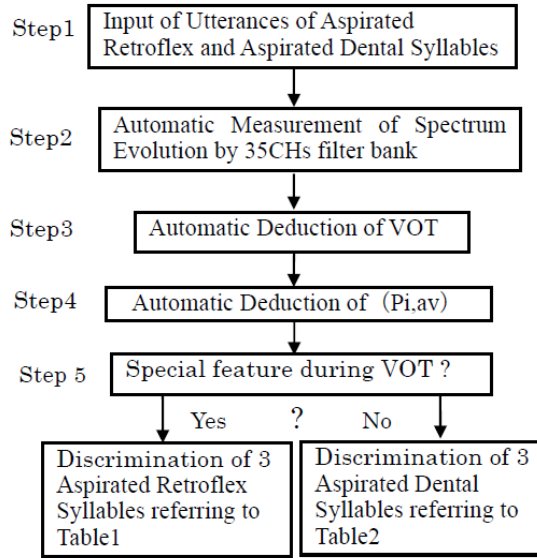


Fig. 7. Discrimination diagram of aspirated retroflex and aspirated dental syllables

In step 4, the average power, $P_{i,av}$, is automatically calculated for each channel during VOT, as described in Subsection 3.2. In step 5, if any distinctive features are found during VOT, they are judged to be aspirated retroflex syllables and discriminated by referring to Table 1.

If there are no distinctive features during VOT, they are judged to be aspirated dental syllables and discriminated referring to Table 2. The $P_{vi,av}$ is automatically calculated for each channel in voiced period, T_{vs} , as described in the Subsection 3.2.

Table 3. Correct judgment rate of aspirated retroflex and dental syllables

	Aspirated retroflex syllables			Aspirated dental syllables		
	cha[tɕ'a]	chi[tɕ'i]	che[tɕ'ɤ]	ca[t'a]	ci[t'i]	ce[t'ɤ]
Correct rate	95%	100%	90%	100%	100%	95%

Table 3 lists the correct judgment rate of aspirated retroflex syllables cha[tɕ'a], chi[tɕ'i], che[tɕ'ɤ] and aspirated dental syllables, ca[t'a], ci[t'i] and ce[t'ɤ] pronounced by 20 Japanese students. All utterances were evaluated to be correct by a listening test involving four native Chinese speakers.

The correct judgment rate of aspirated retroflex syllable cha[tɕ'a] was 95%. One sample was too weak to be correctly detected. The correct judgment rate of retroflex syllable chi[tɕ'i] was the perfect at 100%, and that of aspirated retroflex syllables che[tɕ'ɤ] was the lowest at 90%. One utterance was too weak and another was too strong.

The correct judgment rates of aspirated dental syllables ca[tʰa] and ci[tʰi] were perfect at 100%, and that of aspirated dental syllables ce[tʰʂ] was 95%. One utterance had too little power.

6 Conclusion

We have been studying the instruction of pronunciation of Chinese aspirated sounds, which are generally difficult for Japanese students to perceive and reproduce. We closely examined the spectrograms of uttered sounds by native Chinese speakers and Japanese students and determined the criteria for correct pronunciations of various aspirated sounds [1-5]. We previously developed an automatic system for measuring and calculating the VOT and the power during VOT of student pronunciations [6,7].

In this paper, in order to develop an automatic training system for Chinese pronunciation, we aimed at automatic distinction of the three pairs of aspirated dental and aspirated retroflex syllables ca[tʰa] - cha[tʂʰa], ci[tʰi] - chi[tʂʰi], and ce[tʰʂ] - che[tʂʰʂ]. We automatically calculated the frequency spectrum of the utterance during VOT and voiced periods and extracted the distinctive feature of each utterance. Then we established criteria for automatically discriminating aspirated retroflex and aspirated dental sounds.

We conducted an experiment on automatic discrimination of 20 utterances of Japanese students using our automatic discriminating system. The results of the test showed that the system exhibited an average correct judgment rate for three aspirated retroflex syllables of 90% or more and aspirated dental syllables of 95% or more for the pronunciations evaluated to be correct by native speakers.

The authors appreciate for the financial support by Japan Society for the Promotion of Sciences (JSPS).

References

1. Hoshino, A., Yasuda, A.: Evaluation of Chinese aspiration sounds uttered by Japanese students using VOT and power. *Acoust. Soc. Jpn.* 58(11), 689–695 (2002) (in Japanese)
2. Hoshino, A., Yasuda, A.: The evaluation of Chinese aspiration sounds uttered by Japanese student using VOT and power. In: *IEEE Proceedings of 2003 International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, pp. 472–475 (2003)
3. Hoshino, A., Yasuda, A.: Dependence of correct pronunciation of Chinese aspirated sounds on power during voice onset time. In: *Proceeding of ISCSLP 2004*, Hong Kong, pp. 121–124 (2004)
4. Hoshino, A., Yasuda, A.: Effect of Japanese articulation of stops on pronunciation of Chinese aspirated sounds by Japanese students. In: *Proceeding of ISCSLP 2004*, Hong Kong, pp. 125–128 (2004)
5. Hoshino, A., Yasuda, A.: Evaluation of aspiration sound of Chinese labial and alveolar diphthong uttered by Japanese students using voice onset time and breathing power. In: *Proceeding of ISCSLP 2006*, Singapore, pp. 13–24 (2006)
6. Hoshino, A., Yasuda, A.: Pronunciation Training System for Japanese Students Learning Chinese Aspiration. In: *The 2nd International Conference on Society and Information Technologies (ICSIT)*, Orlando, Florida, USA, pp. 288–293 (2011)

7. Hoshino, A., Yasuda, A.: Pronunciation Training System of Chinese Aspiration for Japanese Students. *Acoustical Science and Technology* 32(4), 154–157 (2011)
8. Hoshino, et al.: *Acoustics 2012*, Nantes, France, pp. 339–344 (April 2012)
9. Kent, R.D., Read, C.: *The Acoustic Analysis of Speech*, pp. 105–109. Singular Publishing Group, Inc., San Diego (1992)
10. Zhu, C.: *Studying Method of the Pronunciation of Chinese Speech for Foreign Students*, pp. 63–71. Yu Wu Publishing Co., China (1997) (in Chinese)

A New Word Language Model Evaluation Metric for Character Based Languages

Peilu Wang, Ruihua Sun, Hai Zhao, and Kai Yu

Institute of Intelligent Human-Machine Interaction
MOE-Microsoft Key Lab. of Intelligent Computing and Intelligent Systems
Department of Computer Science and Engineering
Shanghai Jiao Tong University, 200240, Shanghai, P.R. China
{plwang1990, sun.r.h, zhaohai, kai.yu}@sjtu.edu.cn

Abstract. Perplexity is a widely used measure to evaluate word prediction power of a word-based language model. It can be computed independently and has shown good correlation with word error rate (WER) in speech recognition. However, for character based languages, character error rate (CER) is commonly used instead of WER as the measure for speech recognition, although language model is still word based. Due to the fact that different word segmentation strategies may result in different word vocabulary for the same text corpus, in many cases, word-based perplexity is incompetent to evaluate the combined effect of word segmentation and language model training to predict final CER. In this paper, a new word-based language model evaluation measure is proposed to account for the effect of word segmentation and the goal of predicting CER. Experiments were conducted on Chinese speech recognition. Compared to the traditional word-based perplexity, the new measure is more robust to word segmentation and shows much more consistent correlation with CER in a large vocabulary continuous Chinese speech recognition task.¹

Keywords: language model evaluation, character error rate, perplexity.

1 Introduction

In speech recognition, language model plays an important role. It models the prior probabilities of all possible word sequence that a speech recogniser can deal with. It is independent of acoustic observations and defines the search space of a speech recogniser. In speech recognition, word error rate (WER) is usually used as the ultimate evaluation metric for the whole system. Although WER can also be used to evaluate language model given a fixed acoustic model, it is not convenient to do so because acoustic data is required and decoding is timeconsuming. To conveniently evaluate the quality of an estimated language model, perplexity was proposed and has been the most widely used metric [5]. Perplexity is

¹ This research was partly supported by the Program for Professor of Special Appointment(Eastern Scholar) at Shanghai Institutions of Higher Learning and the China NSFC project No.61222208.

essentially the exponent of the cross entropy between the real word sequence distribution and the estimated word sequence distribution. Its calculation is independent of acoustic data and can be done quickly. More importantly, it was shown that perplexity has good correlation with WER [1,6]. Hence, it has been used for decades to evaluate language model in speech recognition. However, there has also been a long argument about the correlation between perplexity and WER. Previous works showed that the good correlation between perplexity and WER only exists in certain cases [3] and modifications of perplexity has been proposed to improve the correlations in more general cases [3,4,2].

In these studies, different factors are changed to construct different language models, such as corpus size, smoothing algorithm, interpolation weight and so on. Then the correlation between perplexity and WER of all different language models is investigated. However, all the previous works, to our best knowledge, have not explicitly considered the influence of vocabulary on language model training. It may be because that vocabulary is normally fixed before language model training given certain training corpus and consequently does not have remarkable influence. Although this is a common case in word based languages, in character based languages such as Chinese, the influence of vocabulary can not be neglected. Since character based languages are not naturally defined with spaces appearing between words, corpus needs to be segmented to form words before language model training. Different segmentation strategies will generate different word vocabularies with totally different size and components which lead to different probability distribution and final recognition result. We will show in the following chapter that in this situation, perplexity is incompetent to predict the recognition performance.

What's more, for character based languages, character error rate(CER) was used to evaluate the final performance instead of word error rate because character becomes the basic unit while language model is still trained based on word, since word based language model always tends to get a better performance in application. This mismatch makes it harder for perplexity to do an accurate evaluation that perplexity only considers the probability distribution of each word but ignores the information of word itself. For example, it is intuitive that the length of word have relation with the CER because word with more characters will cause more incorrectly recognised characters in CER calculation and this effect will not be recognised by perplexity.

In this paper, traditional word based perplexity is extended to take the effect of vocabulary construction into consideration. Two new evaluation functions are proposed, one is taking the vocabulary size into consideration and the other one is considering the vocabulary size as well as the length of word. Experiments are performed to investigate the correlation between different versions of perplexity and CER, where the segmentaion strategy and word vocabulary are the variable quantities. The result shows that these new measures are more robust and present much more consistent with CER while the influence of word length is not as strong as we thought.

The rest of the paper is arranged as follows. Section 2 reviews traditional word based perplexity and proposes two modified versions for character based languages. Experiments are described in section 3, followed by conclusion.

2 Character Based Perplexity

2.1 Word Based Perplexity and Its Limitation

In natural language processing, it is assumed that the appearance of word in sentences satisfying some specific kind of probability distribution referred to as language model. The model that can best reflect such distribution is called the real model but limited to the calculation ability, it is impossible to achieve this real model in practice. Therefore, the quality of language model is always assessed by quantitatively measuring the difference between the estimated language model and the real model. This can be done by asking how well the estimated model can predict the words generated from the real word distribution. For a given test word sequence $\mathbf{w} = \{w_1, \dots, w_N\}$, where N is the number of words, the perplexity (*ppl*) of the estimated language model $q(\mathbf{w})$ is defined as

$$ppl = 2^{-\frac{1}{N} \log_2 q(\mathbf{w})} = 2^{-\frac{1}{N} \sum_{i=1}^N \log_2 q(w_i|h_i)} \quad (1)$$

where w_i is the i^{th} word of the whole test set \mathbf{w} and $h_i = \{w_1, \dots, w_{i-1}\}$ denotes the word history of w_i . Assuming the real word sequence distribution is $p(\mathbf{w})$, better estimated model $q(\mathbf{w})$ of the unknown distribution $p(\mathbf{w})$ will tend to assign higher probabilities to the test word sequences. Thus, they have lower perplexity, meaning that they are less surprised by the test sample.

Considering $\log_2 q(w_i|h_i)$ represents the bits needed to record the information of word w_i given history h_i , the exponent in equation (1) can be regarded as the number of bits needed per word to represent the test set if the coding scheme used is based on $q(\cdot)$. Low *ppl* means the estimated model requires few bits per word to compress the test set which means the model is more close to the real model.

In most cases, *ppl* calculated by equation (1) works quite well, but when the vocabulary changes, it always tends to behave poorly. Since language model is word based, even for character based language, a word vocabulary is required to determine the set of valid words. Words not appeared in the vocabulary will not be taken into consideration when calculating the *ppl*. The size and composition of the word vocabulary will severely affect *ppls* evaluation. For example, considering two language model LM_A and LM_B , LM_A has only 50 words, and the probability of each word is equal which is $1/50$, and LM_B has 100 words and the probability of each word is also equal for convenience. According to the equation (1), the *ppl* of LM_A is 50 while the *ppl* of LM_B is 100. Although the *ppl* of LM_A is much lower than LM_B , it is likely that, LM_A which contains more words will get a better performance in application due to better coverage of words.

2.2 Character Based Perplexity

Considering the definition of perplexity(ppl), it uses the average bits needed to compress the test set as the criterion to evaluate language model but ignores the vocabulary size. As the example given in last paragraph, it is obviously unfair to compare the number of bits if the two language model have different vocabulary size. Therefore, equation (2) is extended to take the size of vocabulary into consideration. Since this function is designed for conquering problem appearing in character based languages, it is denoted as character-based perplexity ($cppl$) for convenience. The extended function is defined as:

$$cppl = 2^{-\frac{1}{N|V|} \sum_{i=1}^N \log_2 q(w_i|h_i)} \quad (2)$$

where $|V|$ is the number of words of vocabulary V . This is an empirical function that introduces the size of vocabulary as a balancing factor. The language model which has a smaller vocabulary size tends to have larger $q()$ and therefore will get a smaller exponent and smaller ppl . In contrast, in equation (2), the exponent will become larger with smaller vocabulary size which will neutralize the effect of $q(\cdot)$.

What's more, as mentioned before, in character based language, the information of word itself is also an influence factors. Since character based languages are not naturally defined with spaces appearing between words, these words which are decided by segmentation and corpus contains far more possibilities than those in word based languages. For example, on a 310M Chinese text corpus, the size of vocabulary after word segmentation can be more than 1000k! Not only the vocabulary size, the words in vocabulary constructed from different segmentation strategies may also have notable difference. To make it easy to consider this difference, the effect of word length is considered, since it seems intuitive that longer word will cause more error characters if it is incorrectly recognized in speech recognition. The bits needed to transfer the word into equation (2) is further introduced and a refined character-based perplexity, referred to as $cppl_2$ is defined as below:

$$cppl = 2^{-\frac{1}{N|V|} \sum_{i=1}^N \frac{1}{1+\log_2(|w_i|)} \log_2 q(w_i|h_i)} \quad (3)$$

where $|w|$ denotes the number of characters of word w , i.e. word length. This is also an empirical function that considering both of the effect of the vocabulary composition as well as the vocabulary size.

3 Experiments

To investigate the correlation between the new language model evaluation measures and CER, experiments were performed on a large vocabulary Chinese speech recognition task. The acoustic model is a cross-word triphone model trained on about 200 hours of read speech using the minimum phone error (MPE) criterion. It has about 3000 clustered states and an average of 12 Gaussian components per state. The acoustic model was fixed for all experiments. The text

corpus used to train language models were extracted from Weibo² consisting of 42M sentences and 101M characters. A series of trigram language models were then trained during the experiments. The test data for calculating perplexity and CER consists of 2040 sentences, about 20K characters. All these sentences were preprocessed to ensure that they were composed with 6763 simplified Chinese characters and other symbols were filtered. The toolkit to train language model was SRILM[7] and HTK toolkit[8] was used to decode the lattice transcript.

In this experiment, 10 different language models were constructed. Unlike previous works which mostly focused on adjusting the smoothing algorithm or interpolation weight, different language models were generated by utilizing different segmentation strategies in this experiment. To achieve many different segmentation strategies, backward maximal matching(BMM) word segmentation algorithm was used with different vocabularies. These vocabularies was consciously constructed to let the segment result varied obviously, having apparent divergence in word length and vocabulary size to better check the performance of *cppl* and *cppl*₂. The pseudocode generating these vocabularies is shown in Algorithm 1.

These vocabularies are constructed by merging the bigram and trigram in trigram count with high frequency. In our algorithm, if the n-gram($n > 1$) words having high frequency which is represented by the appearance times counted in held out corpus, it is supposed to be a new word and is added to the new vocabulary generated for the next segmentation strategy. The criterion judging high frequency is determined by the input parameter *mc* which represents the number of new word will be added. When the new vocabulary is used for segmentation, many bigrams and trigrams will be recognized as a integrated word which will increase the average word length of the segmented corpus.

The basic information of the 10 segmented corpus to train language models is summarized in Table 1.

Table 1. The average word length and vocabulary size of different language model training corporuses

corpus no	avg word length	vocab size
1	1.0	6k
2	1.44	11k
3	1.62	16k
4	1.72	21k
6	1.79	25k
7	1.85	30k
8	1.89	35k
9	1.93	39k
10	1.97	44k

² Chinese version twitter.

Algorithm 1. Generating segmentation dictionary

```

1: INPUT1 held out corpus hc
2: INPUT2 merge count mc
3: INPUT3 number of generated dictionaries num
4: OUTPUT generated dictionaries vocabs
5: segment hc by characters and get the segmented data sc
6: for  $i=0; i < num; i++$  do
7:   state trigram count tc from sc
8:   for each element e in tc do
9:     if e is trigram then
10:      merge the e to unigram  $e_u$ 
11:      remove e and add  $e_u$  to tc
12:      for each bigram b in e do
13:        if b is in tc then
14:          let  $b.count = e.count$ 
15:        end if
16:      end for
17:    end if
18:    if e is bigram then
19:      merge the e to unigram  $e_u$ 
20:      remove e and add  $e_u$  to tc
21:    end if
22:  end for
23:  sort tc order by count
24:  let  $c = 0$ 
25:  let vocab be the dictionary for new segmentation
26:  for each element e in tc do
27:    if e is merged by trigram or bigram then
28:       $c += 1$ 
29:    end if
30:    if  $c > mc$  then
31:      break
32:    end if
33:    add e to vocab
34:  end for
35:  add vocab to vocabs
36:  segment hc using BMM algorithm with vocab and get the segmented data sc
37: end for

```

3.1 Correlation between CER and ppl

With the trigram language models trained on the 10 different text corpora, normal word-based perplexities were calculated and CERs were generated after full decoding on the acoustic data. The correlation between CER and word-based perplexity *ppl* is shown in figure 1

It can be seen that, there is no positive correlation between CER and *ppl*. To quantify the correlation between different metrics with character error rate, linear correlation coefficient (or Pearson coefficient) was calculated to measure

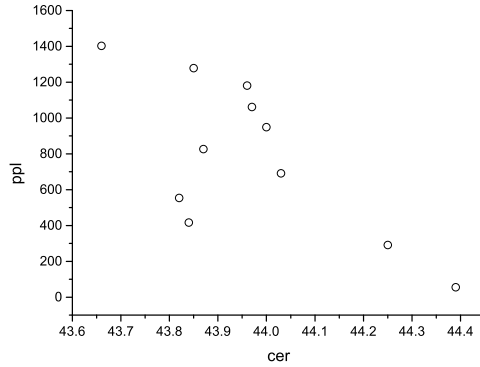


Fig. 1. Correlation of CER and ppl when segmentation strategy varies

the degree of linear correlation. The Linear correlation coefficient of CER and ppl is -0.70 . The coefficient of CER and $\log(ppl)$ is -0.79 . All of the correlation coefficients are negative in this experiment. It is inconsistent with the expectation that CER is positively correlated with ppl .

To further investigated the issue, another experiment has been performed. Here, the segmentation strategy is fixed and the size of corpus to train language model varied from 10M to 100M. The result is shown in figure 2

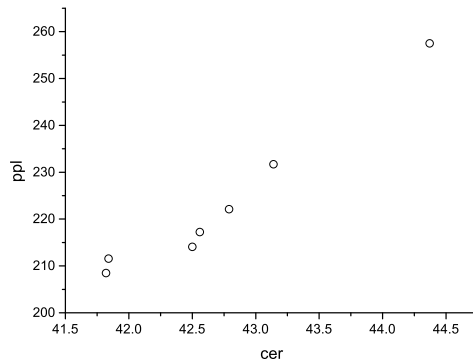


Fig. 2. The correlation of CER and ppl when size of training corpus varies

CER and ppl correlates quite well in this experiment, which is a consistent observation as the previous work on perplexity. From the above two experiments, the correlation between CER and ppl varies from positive to negative which is quite inconsistent, and therefore, we conclude ppl is incompetent for evaluating CER.

3.2 Correlation between CER and *cppl*

The setup of this experiment was same as the previous experiment except equation (2) was used to calculate the *cppl* instead of *ppl*. The correlation between *cppl* and CER when segmentation strategies varies is shown in figure 3 and when the corpus size changes, the correlation is shown in figure 4

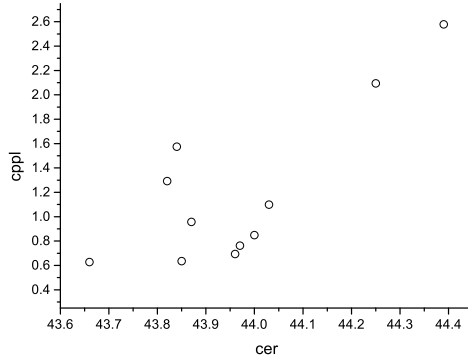


Fig. 3. The relationship between CER and *cppl* when segmentation strategy varies

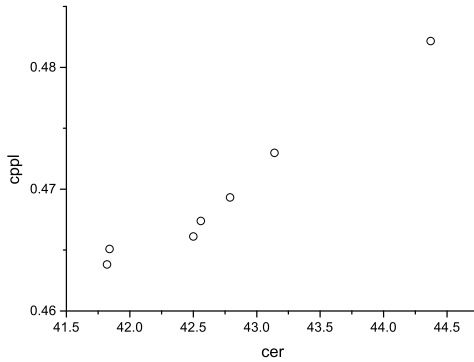


Fig. 4. The relationship between CER and *cppl* when size of training corpus varies

The Linear correlation coefficient in figure 3 is 0.78 which is higher than the absolute value of *ppl* but a little lower than absolute value of $\log(ppl)$ and in figure 4 is 0.97 which is equal to *ppl*. In both figures, *cppl* shows a positive correlation with CER.

Experiment testing the performance of $cppl_2$ which considers the influence of word length was also performed. Equation (3) was used to calculate the $cppl_2$. The correlation result was similar to $cppl$ but with a litter increase in correlation coefficient. The improvement is shown in table 2

Table 2. Linear correlation coefficients comparison

measure	wrd seg vary	corpus size vary
ppl	-0.70	0.97
$cppl$	0.78	0.97
$cppl_2$	0.80	0.98

This comparison showed that considering word length slightly improved the correlation coefficients, but this influence was very tiny compared to the effect caused vocabulary size change.

In the above experiments, it has been shown that perplexity is incompetent predicting language models quality for character based languages. One main reason is that perplexity is not only affected by the probability distribution of language model but also by the scale of vocabulary size. Since it is only the probability distribution deciding the language models performance in speech recognition, the influence of vocabulary size will observably interfere the correlation. Therefore, the proposed metric $cppl$ empirically neutralizing this effect retained inconsistency with character error rate in the two experiments.

The experiment about $cppl_2$ shows that taking the word length into consideration does not have apparent improvement to the evaluation. It infers that word length may not be as important to the correlation as we thought. This is because by our analysis, the influence of vocabulary composition varies is very complex and length only describing a simple physical attribute of a word without reaching its probability attributes or its character element is inadequate to neutralizing the effect caused by vocabulary change. Therefore, our future work will focuses on the further investigation of the influence caused by vocabulary composition, more information and more complex model about the word in vocabulary will be considered.

4 Conclusion

In this paper, perplexity is shown incompetent to predict CER for character based language, since the segmentation strategies which change the vocabulary composition will distinctly affect the evaluation of perplexity. To address this problem, word-based perplexity has been extended. A new metric taking vocabulary size into consideration is proposed. It is shown to successfully neutralize the influence of vocabulary change and is more robust. Length of word in vocabulary is also considered while it is proved having little effect about the final correlation. The main factor about the influence of the vocabulary composition should be further investigated.

References

1. Bahl, L.R., Jelinek, F., Mercer, R.L.: A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2), 179–190 (1983)
2. Chen, S.F., Beeferman, D., Rosenfield, R.: Evaluation metrics for language models (1998)
3. Clarkson, P., Robinson, T., et al.: Towards improved language model evaluation measures. In: *Proc. of EUROSPEECH*, vol. 99, pp. 1927–1930 (1999)
4. Ito, A., Kohda, M., Ostendorf, M.: A new metric for stochastic language model evaluation. In: *Proceedings of the Sixth European Conference on Speech Communication and Technology*, vol. 4, pp. 1591–1594 (1999)
5. Jelinek, F., Mercer, R.L., Bahl, L.R., Baker, J.K.: Perplexity: a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America* 62(S1), S63(1977)
6. Klakow, D., Peters, J.: Testing the correlation of word error rate and perplexity. *Speech Communication* 38(1), 19–28 (2002)
7. Stolcke, A., et al.: Ssrilm—an extensible language modeling toolkit. In: *Proceedings of the International Conference on Spoken Language Processing*, vol. 2, pp. 901–904 (2002)
8. Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: *The htk book* (for htk version 3.2). Cambridge University Engineering Department (2002)

Bidirectional Sequence Labeling via Dual Decomposition

Zhiguo Wang¹, Chengqing Zong¹, and Nianwen Xue²

¹ National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100190

² Computer Science Department, Brandeis University, Waltham, MA 02452
{zgwang, cqzong}@nlpr.ia.ac.cn, xuen@brandeis.edu

Abstract. In this paper, we propose a bidirectional algorithm for sequence labeling to capture the influence of both the left-to-right and the right-to-left directions. We combine the optimization of two unidirectional models from opposite directions via the dual decomposition method to jointly label the input sequence. Experiments on three sequence labeling tasks (Chinese word segmentation, English POS tagging and text chunking) show that our approach can improve the accuracy of sequence labeling tasks when the two unidirectional models individually make highly different predictions.

1 Introduction

Many natural language processing tasks, e.g., POS tagging, text chunking and Chinese word segmentation, can be formulated as a sequence labeling problem. In these tasks, each token in a sequence is assigned a label, and the label assignment of a given token is influenced by the label assignments of the previous tokens. Most sequence labeling models are unidirectional where the inference procedure is performed in one direction only (left to right, or right to left, but not both). As a result, only the influence of one direction is explicitly considered. For many sequence labeling tasks, however, both the left and right contexts can be useful and should be taken into account. For example, consider the POS tagging procedure for the sentence “Would service be voluntary or compulsory?”. The word “service” can either be labeled as a verb or a noun. In a left-to-right model, the POS tag “MD” of the previous word “Would” strongly indicates that “service” should be tagged as verb. However, this is the incorrect answer in the case. In a right-to-left model, the POS tag “VB” of the following word “be” indicates “service” should be a noun, which is the correct answer. This means that a model that accounts for the influence of both the left and right contexts is better.

In recent years, a number of bidirectional sequence labeling models were proposed to exploit the influence of both directions. Liu and Zong (2003) and Shen et al. (2003) improved the tagging accuracy by pairwise combining or voting between the left-to-right and right-to-left taggers. Toutanova et al. (2003) proposed a POS tagging model based on bidirectional dependency networks that make the right context available for a left-to-right model. Tsuruoka and Tsujii (2005) considered all possible decompositions of bidirectional contexts, and chose one that has the highest probability among

different taggers. Shen et al. (2007) extended Tsuruoka and Tsujii (2005) and integrated the inference order selection and classifier training into a single learning framework.

In this paper, we propose a novel approach for bidirectional sequence labeling. We combine the optimization of two unidirectional models from opposite directions to predict agreed labels through the dual decomposition method. We estimated our approach on three sequence labeling tasks for two languages: Chinese word segmentation, English POS tagging and text chunking. Experimental results show that our approach is effective when the two unidirectional models individually make highly different predictions.

2 Unidirectional Approach

Let us denote the input sequence of tokens as $x = x_1 x_2 \dots x_n$, and the label sequence for x as $y = y_1 y_2 \dots y_n$, where y_i (belonging to a label set Y) is the label for the token x_i . For example, in part-of-speech tagging, the input sequence would be the word tokens in a sentence and the output would be POS tags for the word tokens.

The task of sequence labeling is to find the best label sequence \hat{y} for an input sequence x :

$$\hat{y} = \operatorname{argmax}_{y \in Y} p(y|x) \quad (1)$$

Usually, the global probability $p(y|x)$ can be decomposed into products of a sequence of local predictions. For example, in the left-to-right model, the probability is decomposed into:

$$p(y|x) = \prod_{i=1}^n p(y_i|x, y_1 \dots y_{i-1}) \quad (2)$$

where $p(y_i|x, y_1 \dots y_{i-1})$ is the prediction probability of assigning y_i for x_i . Here, we model the prediction probability with the Maximum Entropy (ME) model:

$$p(y_i|x, y_1 \dots y_{i-1}) = \frac{\exp(w \cdot \phi(x, y_1 \dots y_{i-1}, y_i))}{\sum_{y'_i} \exp(w \cdot \phi(x, y_1 \dots y_{i-1}, y'_i))} \quad (3)$$

where $\phi(x, y_1 \dots y_{i-1}, y_i)$ is a feature vector, and w is the weight vector for those features. When given a training set of labeled sequences, we can estimate the model parameter w using the usual way for ME models, i.e., Generalized Iterative Scaling (GIS) or gradient descent methods.

The probability of the current label prediction in Eq (3) is conditioned on label predictions for previous tokens. If we make a first-order Markov assumption, the Viterbi algorithm would be an efficient decoding method. However, Jiang et al., (2008) showed that non-local features are much helpful for POS tagging. Therefore, we design a unidirectional decoding algorithm that uses more than one prediction before the current position.

Algorithm 1 shows the decoding algorithm, which is based on the beam search algorithm. We use two max-heaps to hold the partial label sequences, where *preHeap* maintains a list of N best partial candidates ending at position $i-1$ and *curHeap* maintains a list of N best partial candidates ending at position i . The algorithm initializes the *preHeap* with an empty sequence (line 1). It then traverses the input sequence from left to right, and assigns a label to each token (line 2 to line 13). When processing the i -th token x_i , the algorithm extracts the top partial candidate *item* from *preHeap* (line 6), and tries to extend *item* with each label in the label set \mathbf{Y} . If a label y_i is compatible with *item* (line 9), we build a new partial candidate *item'* by combining y_i with *item* (line 11), calculate the probability of *item'* using E.q. (2) (line 10) and add it to *curHeap* (line 12). When all the input tokens are processed, the best partial candidate in *curHeap* is returned as the final result (line 14).

Algorithm 1. Unidirectional Decoding Algorithm

Input: sequence $x = x_1 \dots x_n$, beam size N
Output: label sequence $y = y_1 y_2 \dots y_n$

- 1: *preHeap* \leftarrow New-Item(*null*)
- 2: **for** $i \leftarrow 1 \dots n$ **do**
- 3: *curHeap* $\leftarrow \emptyset$
- 4: $k \leftarrow 0$
- 5: **while** $|preHeap| > 0$ and $k < N$ **do**
- 6: *item* \leftarrow Pop-Max(*preHeap*)
- 7: $k \leftarrow k + 1$
- 8: **for** y_i in \mathbf{Y} **do**
- 9: **if** IsCompatible(*item*, y_i , x_i) **then**
- 10: $prob \leftarrow$ Eval($i, x, item, y_i$)
- 11: *item'* \leftarrow New-Item(*item*, $y_i, prob$)
- 12: Push(*curHeap*, *item'*)
- 13: *preHeap* \leftarrow *curHeap*
- 14: **return** Pop-Max(*curHeap*)

Although the model and the decoding algorithm are designed for the left-to-right direction, they can be trivially adapted to the right-to-left direction. To train a right-to-left model, we just reverse all the label sequences in the training set before training. For decoding, we reverse the input sequence first, then decode the reversed sequence with the right-to-left model and reverse the label sequence back.

3 Bidirectional Decoding

In this section, we describe how to improve sequence labeling by jointly optimizing the two unidirectional models. We train a left-to-right model and a right-to-left model and then jointly label an input sequence with the two models.

For purposes of clarity, we define some notations first. The label sequence from the left-to-right model is denoted as $l = l_1 l_2 \dots l_n$, and the output from the right-to-left

model is denoted as $r = r_1 r_2 \dots r_n$. For $l = l_1 l_2 \dots l_n$, we define $l(i, t) = 1$ if l_i is assigned with a label $t \in Y$, otherwise $l(i, t) = 0$. Similarly, for $r = r_1 r_2 \dots r_n$, we define $r(i, t) = 1$ if r_i is assigned with a label $t \in Y$, otherwise $r(i, t) = 0$. Therefore, l and r are equal, only if $l(i, t) = r(i, t)$ for all $i \in [1, n]$ and $t \in Y$, otherwise they are unequal.

We expect the two unidirectional models to predict equal results and formulate it as a constraint optimization problem:

$$(\hat{l}, \hat{r}) = \operatorname{argmax}_{l, r} f_1(l) + f_2(r)$$

Such that for all $i \in [1, n]$ and $t \in Y$: $l(i, t) = r(i, t)$

where $f_1(l) = \log p(l|x) = \sum_{i=1}^n \log p(l_i|x, l_1 \dots l_{i-1})$ is a score estimated from the left-to-right model, and $f_2(r) = \log p(r|x)$ is a score estimated from the right-to-left model.

The dual decomposition (a special case of Lagrangian relaxation) method introduced in Rush et al. (2010) is suitable for this problem. Following their method, we solve the primal constraint optimization problem by optimizing the *dual* problem. First, we introduce a vector of Lagrange multiplier $\mu(i, t)$ for each equality constraint: $l(i, t) = r(i, t)$. Then, the Lagrangian is formulated as:

$$L(l, r, \mu) = f_1(l) + f_2(r) + \sum_{i, t} \mu(i, t)(l(i, t) - r(i, t))$$

By grouping the terms that depend on l and r , we rewrite the Lagrangian as

$$L(l, r, \mu) = \left(f_1(l) + \sum_{i, t} \mu(i, t)l(i, t) \right) + \left(f_2(r) - \sum_{i, t} \mu(i, t)r(i, t) \right)$$

Then, the *dual objective* is

$$\begin{aligned} L(\mu) &= \max_{l, r} L(l, r, \mu) \\ &= \max_l \left(f_1(l) + \sum_{i, t} \mu(i, t)l(i, t) \right) \\ &\quad + \max_r \left(f_2(r) - \sum_{i, t} \mu(i, t)r(i, t) \right) \end{aligned}$$

The dual problem is to find the $\min_{\mu} L(\mu)$.

We use the subgradient method (Boyd et al., 2003) to minimize the dual. Following Rush et al. (2010), we define the subgradient of $L(\mu)$ as:

$$\gamma(i, t) = l(i, t) - r(i, t) \quad \text{for all } (i, t).$$

Then, adjust $\mu(i, t)$ as follows:

$$\mu'(i, t) = \mu(i, t) - \delta(l(i, t) - r(i, t))$$

where $\delta > 0$ is a step size.

Algorithm 2. Bidirectional Decoding Algorithm

```

1: Set  $\mu^{(0)}(i, t)=0$ , for all  $i \in [1, n]$  and  $t \in Y$ 
2: for  $k = 1$  to  $K$  do
3:    $\hat{l}^{(k)} \leftarrow \operatorname{argmax}_l (f_1(l) + \sum_{i,t} \mu^{(k-1)}(i, t)l(i, t))$ 
4:    $\hat{r}^{(k)} \leftarrow \operatorname{argmax}_r (f_2(r) - \sum_{i,t} \mu^{(k-1)}(i, t)r(i, t))$ 
5:   if  $l^{(k)}(i, t) = r^{(k)}(i, t)$  for all  $(i, t)$  then
6:     return  $(\hat{l}^{(k)}, \hat{r}^{(k)})$ 
7:   else
8:      $\mu^{(k)}(i, t) = \mu^{(k-1)}(i, t) - \delta (l^{(k)}(i, t) - r^{(k)}(i, t))$ 

```

Algorithm 2 presents the subgradient method to solve the dual problem. The algorithm initializes the Lagrange multiplier values with 0 (line 1) and then iterates many times. At each iteration, the algorithm finds the best $\hat{l}^{(k)}$ and $\hat{r}^{(k)}$ through the left-to-right model (line 3) and the right-to-left model (line 4) individually. If $\hat{l}^{(k)}$ and $\hat{r}^{(k)}$ are equal (line 5), then the algorithm returns the solution (line 6). Otherwise, the algorithm adjusts the Lagrange multiplier values based on the differences between $\hat{l}^{(k)}$ and $\hat{r}^{(k)}$ (line 8). A crucial point is that the argmax problems in line 3 and line 4 can be solved efficiently using the original unidirectional decoding algorithms, because the Lagrange multiplier can be regarded as adjustments for the prediction score $\log p(y_i|x, y_1 \dots y_{i-1})$ of each token. According to the strong duality theorem (Korte and Vygen, 2008), the dual solution is the label sequence we want to get.

4 Experiment

To evaluate the effectiveness of our method, we conducted experiments on three sequence labeling tasks: Chinese word segmentation, English POS tagging and text chunking.

4.1 Tasks and Data Sets

The task of Chinese word segmentation is segmenting a sequence of Chinese characters into words. The character-based model (Xue, 2003) treats segmentation as a sequence labeling task, where each Chinese character is labeled with a tag. We used the tag set used in Wang et al. (2011). We split the Chinese Treebank Version 5.0 (CTB5) with the standard data split: 1-270, 400-1151 as the training set, 301-325 as the development set and 271-300 as the test set.

We split the Penn Wall Street Journal Treebank (WSJ) with the standard data split for POS tagging: sections 0-18 as the training set, sections 19-21 as the development set and sections 22-24 as the test set.

The task of text chunking is to find non-recursive phrases in a sentence. We treat it as a tagging task by converting chunks into tags on tokens. We choose the IOB scheme: each token gets the label B-X if it is the first token in chunk X, the label I-X

if it is not the first token in chunk X, or the label O if it is outside of any chunks. We used the data set from the CoNLL-2000 shared task.

The feature templates for each task are adopted from previous work. For Chinese word segmentation, we use the feature templates provided in Wang et al. (2011). For POS tagging and chunking, we used the feature templates provided in Tsuruoka and Tsujii (2005), excluding those templates containing future predictions.

4.2 Results

We built three systems for each task. The “left-to-right” system and the “right-to-left” system were two unidirectional systems, which trained models and decoded sequences from opposite directions. The “bidirectional” system used these two unidirectional models jointly to decode sequences with Algorithm 2. We trained models for three tasks with the Maximum Entropy model implemented in the OpenNLP toolkit.

We tuned parameters on the development set and finally set the beam size (in Algorithm 1) to $N=20$, the maximum iteration to $K=30$ and the step size to $\delta=0.5$ (in Algorithm 2). The experimental results on the test set are presented in Table 1 and they show that the accuracy of the POS tagging task and the F1 score of the chunking task were improved when using the bidirectional decoding algorithm. However, the Chinese word segmentation task showed no improvement.

Fig. 1 illustrates how the bidirectional de-coding algorithm leads to improvement over unidirectional models when assigning POS tags to the sentence “Would service be voluntary or compulsory?”. In the left-to-right model, the word token “service” is labeled with an erroneous tag “VB”, because the preceding word “Would” is a modal verb that is often followed by a verb. In the right-to-left model, “service” is correctly labeled, because the following word “be” is a verb that is often preceded by nouns. However, the right-to-left model assigns the wrong tag “NN” to the word “compulsory”, presumably because it is the first token in the sequence and “NN” is a more likely tag for the first token. The left-to-right model, on the other hand, assigns the correct label “JJ”. The bidirectional algorithm combines the strengths of both models and assigns the correct tags to all words.

Table 1. Experimental results on the test set

		F1(%)
Chinese Word Segmentation	left-to-right	97.67
	right-to-left	97.55
	bidirectional	97.65
		Accuracy(%)
POS Tagging	left-to-right	96.83
	right-to-left	96.84
	bidirectional	97.15
		F1(%)
Chunking	left-to-right	93.42
	right-to-left	93.37
	bidirectional	93.61

	Would service be voluntary or compulsory ?						
Gold	MD	NN	VB	JJ	CC	JJ	.
Left-to-right:	MD	VB	VB	JJ	CC	JJ	.
Right-to-left:	MD	NN	VB	JJ	CC	NN	.
Bidirectional:	MD	NN	VB	JJ	CC	JJ	.

Fig. 1. A POS tagging example, where the wrong tags are highlighted with red color

4.3 Discussion

To understand the scenarios where the bidirectional decoding algorithm is effective, we analyzed the three tasks in detail. Table 2 presents the total number of tokens in the test set and the number of tokens to which the left-to-right and right-to-left models assigned different labels. We found the number of tokens receiving different labels was low for the Chinese word segmentation task, but high for the English POS tagging and chunking tasks. Combined with the results in Table 1, we can conclude that our algorithm is effective when the two unidirectional models make very different predictions. When the two unidirectional models make the same predictions, even if the predictions are wrong, the bidirectional algorithm can do nothing to correct them.

We also estimated the convergence of the bidirectional decoding algorithm by counting the number of iterations when the two unidirectional models make different predictions. Fig. 2 shows the percentage of sequences where exact solutions are returned versus the number of iterations. We find our algorithm produces exact solutions to over 80% of the sequences within 10 iterations.

Table 2. Differences between the left-to-right and the right-to-left results

	Total Tokens	Inconsistent Tokens
Word Seg.	13,738	48
POS Tagging	129,654	2,384
Chunking	47,377	980

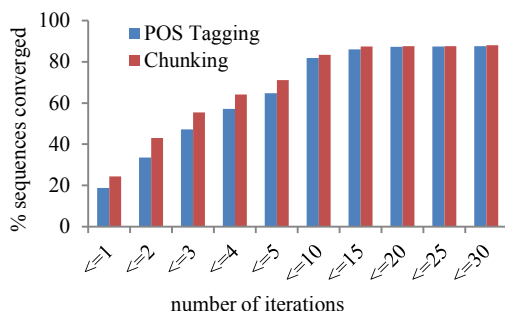


Fig. 2. Convergence of the bidirectional decoding algorithm

5 Conclusion

In this paper, we proposed a bidirectional decoding algorithm for sequence labeling tasks. We use two unidirectional models of opposite directions to jointly label the input sequences via the dual decomposition algorithm. Experiments on three sequence labeling tasks show that our approach improves the performance on sequence labeling tasks when the two unidirectional models makes very different predictions.

Acknowledgments. The research work has been funded by the Hi-Tech Research and Development Program (“863” Program) of China under Grant No. 2011AA01A207, 2012AA011101, and 2012AA011102. This work is also supported in part by the DAPRA via contract HR0011-11-C-0145 entitled “Linguistic Resources for Multilingual Processing”.

References

- Boyd, S., Xiao, L., Mutapcic, A.: Subgradient methods. Lecture notes of EE392o, Stanford University (2003)
- Liu, D., Zong, C.: Utterance Segmentation Using Combined Approach Based on Bi-directional N-gram and Maximum Entropy. In: *ACL 2003 Workshop: The Second SIGHAN Workshop on Chinese Language Processing* (2003)
- Jiang, W., Mi, H., Liu, Q.: Word lattice reranking for Chinese word segmentation and part-of-speech tagging. In: *Coling 2008* (2008)
- Korte, B., Vygen, J.: *Combinatorial optimization: theory and algorithms*. Springer (2008)
- Rush, A.M., Sontag, D., Collins, M., Jaakkola, T.: On dual decomposition and linear programming relaxations for natural language processing. In: *EMNLP 2010* (2010)
- Shen, L., Joshi, A.K.: A SNoW based Supertagger with Application to NP Chunking. In: *ACL 2003* (2003)
- Shen, L., Satta, G., Joshi, A.K.: Guided Learning for Bidirectional Sequence Classification. In: *ACL 2007* (2007)
- Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: *NAACL 2003* (2003)
- Tsuruoka, Y., Tsujii, J.: Bidirectional inference with the easiest-first strategy for tagging sequence data. In: *EMNLP 2005* (2005)
- Wang, Y., Kazama, J., Tsuruoka, Y., Chen, W., Zhang, Y., Torisawa, K.: Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In: *IJCNLP 2011* (2011)
- Xue, N.: Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing* 8(1), 29–48 (2003)

Semantic Analysis of Chinese Prepositional Phrases for Patent Machine Translation

Renfen Hu, Yun Zhu, and Yaohong Jin

Institute of Chinese Information Processing, Beijing Normal University, Beijing, China
bnuhurenfen@126.com, diana_zhupier@hotmail.com,
jinyaohong@bnu.edu.cn

Abstract. In Chinese patent texts, prepositional phrases(PP) are quite long with complicated structures. The correct identification of PP is very important for sentences parsing and reordering in machine translation. However, existing statistical and rule-based methods perform poorly in identifying these phrases because of their unobvious boundaries and special structures. Therefore, we present a method based on semantic analysis. Chinese prepositions are divided into two categories due to their semantic functions, and more contextual features are employed to identify the phrase boundaries and syntax levels. After integrating into a patent MT system, our method has effectively improved the parsing result of source language.

Keywords: Machine translation, Patent, Prepositional phrase, Identification, Syntactic analysis.

1 Introduction

Patent machine translation (MT) is one of the major application fields of MT [1]. However, sentences in Chinese patent texts are known for their complicated structures with multiple verbs and prepositions. It is necessary to make syntactic analysis of source language to deal with the long distance reordering and translation, and the identification of Chinese prepositional phrases (PP) plays an important role in this analysis.

After analyzing sentences of 500 Chinese patent texts, we find that a patent sentence contains approximately 1.9 prepositional phrases (PP) in average, and the average length of each PP is 12.3 Chinese characters, while in news corpus PP contains only 4.9 characters in average [2]. Therefore, the identification of PP in patent texts is concerned in this paper, and we will present a method based on semantic analysis. After fully considering the functions and contextual information of PPs, we classify all prepositions into two semantic categories, and define some contextual features for each. In the analysis, phrase boundaries and syntax level are successively determined based on search algorithm and semantic rules. As a result, the correct identification of PPs can help to achieve better results in predicate identification and syntactic reordering.

After integrating our method into an online patent MT system running in SIPO (State Intellectual Property Office of People's Republic of China)¹, we take a closed test and an open test. The result shows that our method has improved the performance of patent translation.

After a discussion of related work in section 2, we will introduce the semantic features in section 3. Section 4 presents the semantic analysis of PP and the processing steps, and section 5 gives the experiment and evaluation. Finally we draw some conclusions in section 6.

2 Related Work

Chinese prepositional phrase differs a lot from English in locations and functions. For this reason, the identification of PP becomes a procedure of crucial importance in Chinese-English machine translation.

Researchers find that PP mainly served as attribute, adverbial, complement or other adjuncts, and verbs in PP cannot be core predicates. Thus with the identification of PPs, we can narrow down the list of probable predicates, and the syntactic analysis can also be greatly simplified [2][3].

In recent years, a number of statistical methods have been proposed to make text chunking, an intermediate step towards full parsing. PPs, as well as other type of phrases, are identified in this analysis with statistical models such as HMM, maximum entropy, SVM and so on [4]. In order to deal specifically with the identification of PPs, linguistic rules are integrated into the statistical methods, which have greatly improved the identification result [5].

However, existing methods perform poorly in identifying PPs in patent sentences. As mentioned above, PPs in Chinese patent texts are often quite long with complicated structures, which may contain nested phrases, or even clauses. In addition, phrase boundaries are often omitted. While in most statistical systems, phrase boundaries are determined by probabilities depending on features of no more than 5 words. Moreover, in both statistical models and linguistic rules, words' features and contextual information are very limited, including only word collocations and part of speeches. On account of data sparseness and limited features, it turns out to be extremely difficult for existing systems to perform well in PP identification in patent corpus.

To solve this problem, we will describe an approach based on semantic analysis, which employs more contextual information and features of prepositions, including their semantic categories, functions, positions, collocations, ambiguities and so on. With the identification of phrase boundaries and their syntax levels, we can parse and reorder a sentence more explicitly.

¹ <http://c2e.cnpat.com.cn/sesame.aspx>

3 Semantic Features

3.1 Semantic Categories

One of the important differences between Chinese and English is the function of prepositions. In this part, we will define two semantic categories to draw a clear distinction of them.

In the view of semantics, sentences are composed of propositions and arguments, rather than phrases in syntactic structures. For example, semantic roles are used to make shallow semantic analysis in Proposition Bank [6]. Sentences are annotated with two types of roles, thematic and adjunct. Thematic roles mainly refer to the action or state described by a sentence's predicate, such as agent, patient and experiencer, while adjunct roles represent auxiliary information which is not structurally dispensable in a sentence, such as time, location and manner.

English prepositions mainly introduce adjunct roles, while Chinese prepositions introduce both two types of roles. As shown in table 1, Chinese prepositions can be classified into two categories according to the semantic roles² they introduce.

Table 1. Semantic categories of Chinese prepositions

Semantic Category	Introduced Roles	Example Prepositions
SC0	Thematic roles, such as agent, patient, theme, etc.	把, 将, 对, 由
SC1	Adjunct roles, such as time, location, manner, etc.	在, 通过, 除了, 根据

SC is the abbreviation of Semantic Category. In our knowledge base, 15 Chinese prepositions are labeled as SC[0], and 110 prepositions as SC[1].

3.2 Word Collocations

To identify a phase in a sentence, we need to determine the left and right boundaries. As to Chinese prepositional phrase, the left boundary is the preposition, while the right boundary strongly depends on word collocations. We note that SC0 and SC1 are collocated with different components in sentences, and these are important features for the identification of PPs.

SC0s are special Chinese prepositions which are used to emphasize a part of the sentence, or to make nuance of the meaning by changing the word order. Each SC0 must appear together with a predicate. As shown in sentence 1, 由 and 把 are two Chinese SC0 prepositions. 由 is collocated with 激活, while 把 is collocated with 固定. Thus the predicates in a sentence can help to determine the right boundary of PPs.

² Semantic roles mentioned in this paper are from PropBank, a corpus of text annotated with information about basic semantic propositions. <http://verbs.colorado.edu/~mpalmer/projects/ace.html>.

Sentence 1. 一种由紫外线激活的粘合剂把传感器壳体固定在中支架上。(An adhesive activated by ultraviolet secures the sensor housing to the middle bracket.)

Similar to English prepositions, SC1 don't collocate with predicates. They are used independently or in collocation with postpositions. The prepositional phrases beginning with SC1 can be modifiers of either predicates or noun phrases. In sentence 2, 在下面的酰化纤维素树脂中 and 按照程序 are two PPs beginning with SC1s. We can note that 在 is collocated with postposition 中, while 按照 occurs independently. In sentence 3, there is no postposition after 通过, but the conjunction 来 can suggest the right boundary.

Sentence 2. 在下面的酰化纤维素树脂中, 按照程序详细地描述适用于本发明的处理酰化纤维素膜的方法等。(In the following cellulose acylate resins, methods for processing a cellulose acylate film, etc. suitably used for the present invention will be described in detail following the procedures.)

Sentence 3. 所述共享可以通过扩展频谱数字调制来实现。(Such sharing can be achieved through spread spectrum digital modulation.)

3.3 Verb Valency

Many linguistic theories proposed that a verbal predicate and its arguments can form a predicate-argument structure, in which the arguments help to complete the meaning of the predicate [7][8]. Verb valency(VV) refers to the number of arguments in the structure, and it is an important feature to help us identify SC0 prepositions in sentences.

Sentence 4. Jane sent me a letter.

Sentence 5. Tom hits Bob.

In sentence 4, *sent* is a predicate with 3 arguments (*Jane, me, letter*), so its valency is 3. In sentence 5, *hit* has only 2 arguments(*Tom, Bob*), so its valency is 2.

In the knowledge base, we label the verb valency for each verb as VV[1], VV[2] or VV[3]. This value has played an important role in the identification of PPs, which will be discussed in detail in the following section.

3.4 Syntax Level

We have stated that in patent texts, sentences often contain nested phrases. In this case, the syntax levels of PPs must be distinguished so as to find the correct right boundary for each preposition. Here we define a LEVEL value for PPs according to their node locations in the syntax tree. In our method, the syntax level of a PP is as same as its preposition's. However, the LEVEL values of SC0 and SC1 depend on different factors.

As to PPs beginning with SC0 prepositions, the LEVEL value is determined by the parent node of the PP. we define a PP as LEVEL[1] if it is a child node of S(sentence), as LEVEL[2] if it is a child node of NP.

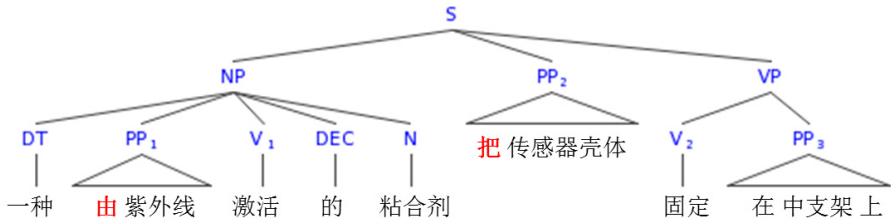


Fig. 1. The syntax tree of Sentence 2

Fig. 1 presents a syntax tree of sentence 1. We can note that PP₂(把传感器壳体) appears independently in the sentence, while PP₁(由紫外线激活) is nested in a NP(noun phrase). Base on this definition, we can give a syntax level analysis shown in table 2.

Table 2. Syntax level analysis of PPs beginning with SC0s

LEVEL	PP	Parent Node	LB*	RB* Information
[1]	把传感器壳体	S	把/SC0	V
[2]	由紫外线	NP	由/SC0	V+的

*LB: left boundary; RB: right boundary

As to PPs beginning with SC1 prepositions, we define the LEVEL value as follows. Given two PPs, PP_i and PP_j, if PP_i is nested in PP_j, then PP_i is LEVEL[2], PP_j is LEVEL[1].

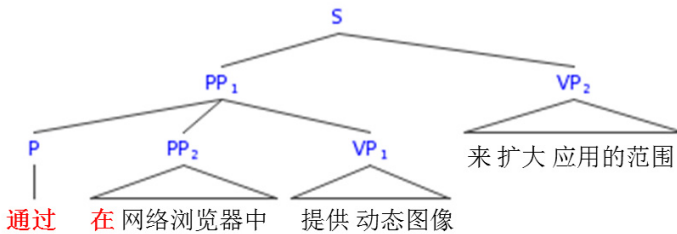


Fig. 2. The syntax tree of a patent sentence

Fig. 2 presents a sentence with two SC1 prepositions. We can also give a syntax level analysis of the PPs in table 3.

Table 3. Syntax level analysis of PPs beginning with SC1s

LEVEL	PP	LB	RB Information
[1]	通过网络浏览器中提供动态图像	通过/SC1	来
[2]	在网络浏览器中	在/SC1	中

4 Semantic Analysis of Chinese Prepositional Phrases

Given a patent sentence $S=W_1, W_2, W_3 \dots W_{n-2}, W_{n-1}, W_n$, let W_i be a preposition, i.e. the LB(left boundary), and W_j be the RB(right boundary). Therefore, to identify a PP, we need to determine three parameters, W_i, W_j and the syntax level of the phrase. In this section, we will discuss the semantic analysis of PPs beginning with SC0 and SC1 separately.

4.1 The Identification of PPs Beginning with SC0

Chinese PP beginning with SC0 has no obvious right boundary(RB). However, as we mentioned above, each SC0 must appear together with a predicate, so the valency and location of verbs have played important roles in the identification.

Table 4. Basic collocations of SC0s and verbs

W_i (LB)	W_j (RB)	W_{j+1}	W_{j+2}	LEVEL	Example Sentence
SC0	-	Verb & VV[2]	PU*	1	硬件结构也仅由一块半导体芯片实现。
SC0	-	Verb & VV[2]	在,到,给,成,为,至于	1	通信模块将数据发送到计算机系统。
SC0	-	Verb & VV[2]	的	2	一种与抗原蛋白靶位互补的肽
SC0	-	Verb & VV[3]	!PU&!的	1	工作人员把药物注入容器。

* PU refers to Chinese punctuations, such as , , ; , and 。

As shown in table 4, W_i can be determined by the collocation of SC0 and a verb(W_{j+1}), and there are four basic types of collocations, in which the right boundary(RB) and syntax level of a PP are dependent on verb valency of W_{j+1} and its location. It is important that some collocations are not applicable to all SC0s. For example, 与 cannot collocate with a structure of Verb&VV[2]+在,到,给,成,为,至于. These details are considered in our semantic rules.

After fully considering the contextual information, we have made 43 rules for the identification, including 2 steps. In step 1, PPs are identified as LEVEL 1. In step 2, the phrases are identified as LEVEL 2. In our model, rules are circularly matched until the system has nothing new to output. If a phrase is given multiple LEVEL values, take the last one. Here Sentence 1 is taken as an example to illustrate the identification process.

Sentence 1. 一种由紫外线激活的粘合剂把传感器壳体固定在中支架上。(An adhesive activated by ultraviolet secures the sensor housing to the middle bracket.)

Step 1. 由 and 把 are both identified as LEVEL[1] by matching Rule 1.

Step 2. 由 is identified as LEVEL[2] by matching Rule 2.

Rule 1. $(0)SC[0]+(f)(m)Verb&VV[2]]+(m+1)CHN[在,到,给,成,为,至于]=> LB(0)+RB(m-1)+PUT(LEVEL,1)$

Rule 2. $(0)SC[0]+(f)(m)Verb&VV[2]]+(m+1)CHN[的]=> LB(0)+RB(m-1)+PUT(LEVEL,2)$

After the analysis module, 由紫外线 is identified as a PP in LEVEL[2], and 把传感器壳体 is identified as PP in LEVEL[1].

In addition to the collocations of SC0s and verbs, we also find other information that can help to determine the level of SC0. For example, if a SC0 appears behind a SC1 preposition, numeral, quantifier or pronoun, we can put it in LEVEL[2].

4.2 The Identification of PPs Beginning with SC1s

As we discussed in section 3, SC1 prepositions mainly introduce adjunct roles, thus in most cases the PPs beginning with SC1s are modifiers of predicates or NPs. With SC1 as a certain left boundary, we need to determine the right boundary and syntax level of the PP. Note that the LEVEL value is given only when PPs are nested, so not all PPs has LEVEL values. After analyzing 500 Chinese patent texts, we have found different contextual information and collocations for SC1 prepositions. Table 5 shows some basic identification patterns, and SC1 varies in different collocations.

Table 5. Basic patterns for identification of PPs beginning with SC1s

$W_i(LB)$	$W_j(RB)$	W_{j+1}	Example Sentence
SC1	Postposition	-	对于HTTP摘要而言将是这种情况。
SC1	-	以, 来, 而	可使用DNS来供给任意网络服务。
SC1	-	PU	为了找到突发脉冲的最优定时,
SC1	-	Predicate	所捕捉的图像对于该设备呈现白色。
SC1	-	SC0	其根据场景光源对数据进行处理。

Based on above conclusions, we have developed a 3-Step identification model. In Step 1, LBs and RBs are generated in the positions of SC1s and postpositions. In Step 2, we check if the LB or RB of current node is nested in another PP with search algorithm, and give LEVEL value to the LB or RB of nested phrase through 12 semantic rules. In Step 3, all PPs are generated and LEVEL values are given to the nested phrases. Here we take sentence 6 as an example to illustrate the semantic analysis.

Sentence 6. 根据本发明的示例性实施例, 可通过在网络浏览器中提供动态图像来扩大UI显示方法的应用范围。 (According to exemplary embodiments of the present invention, by providing a dynamic image in a web browser, the range of applications of the UI display method can be enlarged.)

Step 1. 根据, 通过, 在 are identified as LBs, and 中 is identified as a RB.

Step 2. 在 is identified as LB in LEVEL[2], 中 is identified as RB in LEVEL[2].

Step 3. By matching the following 3 rules, the system generates three PPs, PP1(根据本发明的示例性实施例), PP2(通过在网络浏览器中提供动态图像) and PP3(在网络浏览器中). PP3 is given LEVEL[2].

Rule 3. $(0)LB\&CHN[根据]+(f)(m)CHN[,]+(f)(0,m)!Verb=>RB(m-1)+PP(0,m-1)$

Rule 4. $(0)LB\&CHN[通过, 利用, 采用, 使用, 用]+(f)(m)CHN[以, 而, 来]\Rightarrow RB(m-1)+PP(0, m-1)$

Rule 5. $(0)LB\&LEVEL[2]+(f)(m)RB\&LEVEL[2]\Rightarrow PP(0, m)+PUT(LEVEL, 2)$

5 Experiment and Evaluation

The experiment takes 500 authentic patent texts provided by SIPO (State Intellectual Property Office of China) as the training set. The evaluation will use the development data for the NTCIR-9 Patent Machine Translation Pilot Task³, containing 2,000 bilingual Chinese-English sentence pairs.

After integrating the method into a Chinese-English patent machine translation system [9], we take a closed test on training set, and an open test on evaluation set. The precision and recall are calculated for both two tests to evaluate the identification of PPs after semantic analysis. Necessarily, only when the LB, RB and syntax level of a PP are all correctly identified, we count it as a correct identification. In the open test, BLEU score[10] is also employed to evaluate the translation performance. Table 6 shows the result of the closed test.

Table 6. Experiment Result on the Training Set

	Precision(%)	Recall(%)
PP (SC0)	90.91	84.51
PP (SC1)	90.71	88.77

We can note that the recall is lower than precision for both two types of PPs. Two reasons can account for this phenomenon. (1) The preposition is not recognized as a left boundary due to mistakes of segmentation and word sense disambiguation. For example, *对调焦误差信号* is a PP beginning with *对*(SC0), but *对调* is segmented as a word. In the sentence *顾客将编码游戏卡插入其内*, *将* is identified as a verb modifier, not a preposition. (2) In this system, we make strict conditions for the generation of PPs, which might also result in a lower recall.

In the open test, comparison is made as shown in table 7. RB-MT is the baseline system running on SIPO. HYBRID-MT is the system integrated with our semantic analysis. Google is an online statistical MT system, the identification result of which is inferred from its translation result, so we count its identification as correct when the LB and RB are identified correctly, regardless of the syntax level and reordering result.

Table 7. Compared result of PP identification in the open test

	Precision (%)		Recall (%)		F-score (%)	
	PP(SC0)	PP(SC1)	PP(SC0)	PP(SC1)	PP(SC0)	PP(SC1)
RB-MT	71.23	82.51	62.02	74.30	66.31	78.19
HYBRID-MT	88.11	94.09	75.90	89.25	81.55	91.61
GOOGLE	60.71	76.44	51.20	68.22	55.56	72.10

³ <http://research.nii.ac.jp/ntcir/ntcir-9/data.html>

The result of the open test shows that the semantic analysis has effectively improved the identification result of Chinese PPs, and Google performs poorly in this test. It is mainly because statistical methods face difficulties in determining the RB of a long phrase, and technical texts(including patent texts) account for a fairly low proportion in the training bilingual corpus. Thus, our method is advantageous in processing technical texts with long and complicated sentences. In addition, we find that the identification result of PPs with SC1 is generally better than PPs with SC0. According to statistics, about 40% PPs with SC1 have postpositions as the certain right boundaries, while PPs with SC0 does not have any obvious right boundaries, the identification of which mainly depends on contextual information. After calculating the precision and recall, we give the BLEU-4 score of the three systems shown in table 8.

Table 8. The BLEU scores of three MT system

System	BLEU-4
RB-MT	0.1997
HYBRID-MT	0.2233
GOOGLE	0.3076

From table 8, we can see that after integrating the semantic analysis, the BLEU score has increased by 11.82% from 0.1997 to 0.2233. However, the BLEU scores of three systems all not very high, the highest is 0.3076 of Google. It is mainly because the corpus domain is not limited, unknown terms or entities may result in a bad translation performance, and in BLEU-4 evaluation, sentence will be given a score of 0 if it does not have at least one 4-gram match. Besides, we need to note that Google performs better in word selection, so our system needs to improve this module urgently.

After the experiment, we also make analysis of the identification errors and summarize 5 problems that need to be solved in the future. (1) A sentence may contain multiple verbs, which would interfere with the semantic rules; (2) PPs across comma and PPs of nesting level ≥ 3 have not been considered yet; (3) There are labeling mistakes in the knowledge base that needs a careful review; (4) Preprocessing module (word segmentation and sense disambiguation) as we mentioned above also needs to be improved; (5) Our method is strongly dependent on the completeness of rules, which still need to be complemented.

6 Conclusion

To deal with the identification of Chinese prepositional phrases in patent sentences, we present a method based on semantic analysis, and integrate it into a source language parser for patent machine translation.

By identifying PPs of two semantic categories, our method has enhanced the performance of patent machine translation. In the future, the rule set and knowledge base need to be improved, as well as the other analysis modules in the MT system.

Furthermore, our identification method can be extended to language parsing of technical texts in other fields.

Acknowledgment. The authors are grateful to *Multi-level Knowledge Representation of Mass Texts and Chinese Language Understanding System* (National 863 Program, No.2012AA011104) and the Fundamental Research Funds of Central Universities for financial support.

References

1. Jin, Y., Liu, Z.: Improving Chinese-English Patent Machine Translation Using Sentence Segmentation. In: 7th International Conference on Natural Language Processing and Knowledge Engineering, Tokushima, pp. 620–625 (2011)
2. Gan, J., Huang, D.: Automatic Identification of Chinese Prepositional Phrase. *Chinese Information* 19, 17–23 (2005) (in Chinese)
3. Yin, L., Yao, T., Zhang, D., Li, F.: A Hybrid Approach of Chinese Syntactic and Semantic Analysis. *Chinese Information* 04, 45–51 (2002) (in Chinese)
4. Yu, J.: The Automatic Identification of Chinese Prepositional Phrase based on Maximum Entropy. Dalian University of Technology (2006) (in Chinese)
5. Lu, C., Xu, H., Wang, Y.: Research on the Identification of Chinese Prepositional Phrase based on Semantic Analysis. *Computers and Telecommunications* 03, 46–48 (2012) (in Chinese)
6. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics* 31, 71–106 (2005)
7. Gildea, D., Palmer, M.: The Necessity of Parsing for Predicate Argument Recognition. In: Proceedings of the 40th Meeting of the Association for Computational Linguistics, Philadelphia, pp. 239–246 (2002)
8. Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Schasberger, B.: The Penn Treebank: annotating predicate argument structure. In: Proceedings of the Workshop on Human Language Technology, Plainsboro, pp. 114–119 (1994)
9. Wang, D.: Chinese to English automatic patent machine translation at SIPO. *World Patent Information* 31, 137–139 (2009)
10. Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176 (W0109-022), IBM Research Report

Massive Scientific Paper Mining: Modeling, Design and Implementation

Yang Zhou, Shufan Ji, and Ke Xu

State Key Lab. of Software Development Environment
Beihang University, Beijing, 100191, P.R. China

buaazhouyang@foxmail.com, jishufan@buaa.edu.cn, kexu@nlsde.buaa.edu.cn

Abstract. With dramatic increasing of scientific research papers, scientific paper mining systems have become more popular for efficient paper retrieval and analysis. However, existing keyword based search engines, language or topic model based mining systems cannot provide customized queries according to various user requirements. Hence, in this paper, we are motivated to propose a novel TAIL (Time-Author-Institute-Literature) model to capture the relationships among literature, authors, institutes and time stamps. Based on the TAIL model, we implement the Massive Scientific Paper Mining (MSPM) system and set up a B/S (Browser/Server) structure for web services. The evaluation results on large real data show that our MSPM system could deliver desirable mining results, providing valuable data supports for scientific research cooperations.

1 Introduction

Scientific papers, which deliver the latest scientific research progress and achievement, are of great importance for scientific researchers to share ideas, find interested topics, identify potential research directions, and evaluate academic achievement. Nowadays, there are more than 5 million papers published each year, with an annual increase of 7% - 8%, which makes it impossible for researchers to "manually" track all relevant papers of their interested topics from such massive datasets. Therefore, scientific paper mining has been proposed for efficient paper retrieval and analysis.

Language model [1] is one of the most straightforward methods for paper retrieval, hence is widely adopted in scientific paper mining systems. Nowadays, there exist quite a few scientific search and analysis engines for scientific paper mining, such as Google Scholar [2] and Microsoft Academic Search [3]. However, those engines are usually limited to "keyword" search, which cannot satisfy various query demands. Even with the same keyword query, different users may have different expectations. For example, a user might want to get the most related papers that exactly contain the keyword, while another user might want to search for the papers under the topic about the keyword. However, the "keyword" search cannot capture the difference.

As language models usually search papers based on keyword frequency, ignoring the relationships among synonyms and the topics/themes of papers, topic models are proposed to effectively associate keywords and topics. Latent Dirichlet Allocation (LDA) [4] initiates the study of topic models. Then many researchers extend the topic models in different perspectives. Steyvers et al. [5] built probabilistic author-topic models to analyze the relationships between authors and topics. Wang et al. [6] introduced a topic-over-time (TOT) model to capture the time's effect on topic trend. More recently, Tang et al. [7] propose a patent mining method with a combination of the topic model and the language model. However, this work employs the product of the two models' values, thus users cannot balance the tradeoff degree of the two models according to their query expectations.

To satisfy various query expectations, in this paper, we propose a novel TAIL(Time-Author-Institute-Literature) model to capture the relationships among literature, authors, institutes and time stamps. The TAIL model is a combination of three models: Customized Model (CM), Author Model (AM), Institute Model (IM). The CM is a tradeoff-balanced combination of language model and probabilistic topic model, which could deliver various customized paper queries; while AM and IM could identify authors/institutes that are specialties at some hot research topics, as well as generate hot topic lists that are being studied by the authors/institutes, providing valuable data supports for scientific research cooperations. Based on the TAIL model, we implement the Massive Scientific Paper Mining (MSPM) system and set up a B/S (Browser/Server) structure for web services. The evaluation results on large real data show that our MSPM system could deliver desirable mining results for various user query expectations.

The rest of this paper is organized as follows. Section 2 introduces some preliminaries of this paper. Section 3 proposes the TAIL model and Section 4 introduces the implementation structure of TAIL-based MSPM system. We evaluate the performance of MSPM system in Section 5, and draw conclusions in Section 6.

2 Preliminaries

Here, we firstly define some notions that we will use in the paper, and then introduce the language model [1] and topic models [4][6] that we adopted.

2.1 Dictionary

The valid information of the papers is processed to generate multiple dictionaries, including the word dictionary W , the author dictionary A and the institute dictionary I . The elements in each dictionary are distinct. The notions relevant to those dictionaries are defined in Table 1.

Table 1. Notation List

Notation	Description
N_D	The number of papers.
N_W	The number of distinct words in all papers.
N_Z	The number of topics in the topic model.
N_A	The number of distinct authors in all papers.
N_I	The number of distinct institutes in all papers.
i, j, k, l, m	The indexes of the papers, the words, the topics, the authors and the institutes, respectively ($i = 1, 2, \dots, N_D$)($j = 1, 2, \dots, N_W$)($k = 1, 2, \dots, N_Z$)($l = 1, 2, \dots, N_A$)($m = 1, 2, \dots, N_I$).
$\mathbf{D} = \{d_1, d_2, \dots, d_{N_D}\}$	The set of papers, where d_i refers to the i -th paper.
$\mathbf{W} = \{w_1, w_2, \dots, w_{N_W}\}$	The set of distinct words, where w_j refers to the j -th word.
$\mathbf{Z} = \{z_1, z_2, \dots, z_{N_Z}\}$	The set of topics, where z_k refers to the k -th topic.
$\mathbf{A} = \{a_1, a_2, \dots, a_{N_A}\}$	The set of authors, where a_l refers to the l -th author.
$\mathbf{I} = \{i_1, i_2, \dots, i_{N_I}\}$	The set of institutes, where i_m refers to the m -th institute.
N_i	The number of distinct words in the i -th paper.
N^j	The frequency of the j -th word in all the papers.
N_i^j	The frequency of the j -th word in the i -th paper.
n_{ik}	The frequency of the k -th topic in the i -th paper.
n_{kj}	The frequency of the j -th word assigned to the k -th topic.
ξ	Customized factor for the combination of language model and topic model.

2.2 Language Model

The language model is usually associated with a document in a collection. With a query Q as input, retrieved documents are ranked based on the probability that the document’s language model would generate the terms of the query. According to the language model in [1], given a set of papers and a keyword, the relevance between the keyword and the specific paper can be calculated by Eq. (1).

$$\begin{aligned}
 p_{lm}(w_j|d_i) &= \frac{N_i}{N_i + \sigma} \cdot \frac{N_i^j}{N_i} + \left(1 - \frac{N_i}{N_i + \sigma}\right) \cdot \frac{N^j}{N_W} \\
 &= \frac{N_i^j}{N_i + \sigma} + \left(1 - \frac{N_i}{N_i + \sigma}\right) \cdot \frac{N^j}{N_W}
 \end{aligned}
 \tag{1}$$

where N_i is the number of distinct words in the i -th paper, N_i^j is the frequency of the j -th word in the i -th paper, N^j is the frequency of the j -th word in all papers, N_W is the number of distinct words in all papers. σ is the Dirichlet smoothing factor and its value is set according to the average length of the papers in the database [1].

Generally, a query q typed in by users is often composed of multiple single words. Then the probability of a specific paper d_j generating a query q can be calculated by Eq. (2).

$$p_{lm}(q|d_i) = \prod_{\{w_j|w_j \in q\}} p_{lm}(w_j|d_i) \quad (2)$$

2.3 Topic Model

Probabilistic topic models are important tools for scientific paper mining, which identify the latent topics/themes of massive unstructured documents. In topic models, papers can be seen as random mixtures over various topics, each of which can be characterized by a distribution over words. According to the LDA model [4], the paper-topic distribution and the topic-word distribution, Θ and Φ , can be estimated by Eq. (3) and (4).

$$\theta_{ik} = \frac{n_{ik} + \alpha}{\sum_{k=1}^{N_Z} (n_{ik} + \alpha)} \quad (3)$$

$$\varphi_{kj} = \frac{n_{kj} + \beta}{\sum_{j=1}^{N_W} (n_{kj} + \beta)} \quad (4)$$

where n_{ik} is the frequency of the k -th topic in the i -th paper, n_{kj} is the frequency of the j -th word assigned to the k -th topic, α and β are the hyper parameters in the LDA model.

After getting Θ and Φ , we can derive the relevance between a word and a paper by Eq. (5).

$$p_{lda}(w_j|d_i) = \varphi_{\hat{k}j} \cdot \theta_{i\hat{k}} \quad (5)$$

where

$$\hat{k} = \arg \max_k \varphi_{kj} \quad (6)$$

Then the relevance between a query q and a specific paper can be derived by Eq. (7).

$$p_{lda}(q|d_i) = \prod_{\{w_j|w_j \in q\}} p_{lda}(w_j|d_i) \quad (7)$$

To capture the effect of time on the trend of the topics, we adopted TOT [6] into our model. With TOT, the θ_{ik} and φ_{kj} defined in Eq. (3) and (4) become the definitions in Eq. (8) and (9).

$$\theta'_{ik} = \frac{n_{ik}^t + \alpha + \tau(n_{ik}^{t-1} + \alpha)}{\sum_{k=1}^{N_Z} (n_{ik}^t + \alpha) + \tau(\sum_{k=1}^{N_Z} (n_{ik}^{t-1} + \alpha))} \quad (8)$$

$$\varphi'_{kj} = \frac{n_{kj}^t + \beta + \tau(n_{kj}^{t-1} + \beta)}{\sum_{j=1}^{N_W} (n_{kj}^t + \beta) + \tau(\sum_{j=1}^{N_W} (n_{kj}^{t-1} + \beta))} \quad (9)$$

where the superscript t refers to the values at time t , and τ is the parameter that controls the effect of the values in the previous time on that of the current time.

3 TAIL Model

In this section, we will introduce the TAIL Model that contains the Customized Model (CM), the Author Model (AM), and the Institute Model (IM). Our proposed TAIL Model could well capture the correlation of topics, time stamps, authors, institutes and literatures.

3.1 Customized Model

As the language models are usually limited to keyword search, we combine the topic model with the language model for topic/theme related queries. Different from the combined models in [7], which employ the product of language model value and topic model value, we define a customized factor $\xi \in [0, 1]$ to balance the tradeoff of the two models. Users could set different ξ values to balance the tradeoff degree, according to their special query requirements. Thus, the relevance between a keyword and a paper under CM is defined in Eq. (10).

$$p_{cm}(w_j|d_i) = \xi \cdot p_{lm}(w_j|d_i) + (1 - \xi) \cdot p_{lda}(w_j|d_i) \quad (10)$$

It should be noted that the language model and topic model are the special cases of CM where $\xi = 1$ and $\xi = 0$, respectively. Similarly, the relevance between a query and a paper is defined in Eq. (11).

$$p_{cm}(q|d_i) = \prod_{\{w_j|w_j \in q\}} p_{cm}(w_j|d_i) \quad (11)$$

3.2 Author Model and Institute Model

To capture the authors' and the institutes' expertise on specific research areas, we propose model-based analysis of authors and institutes by the Author Model (AM) and the Institute Model (IM), respectively.

To get the ranking list of authors/institutes at some hot research topics, we should measure the relevance score of an author/institute and the papers under specific topics, as well as generate hot topics associated with each author/institute, indicating which topics are the author's/institute's specialties. The methodologies of these two models are similarly summarized as follows.

First, the author/institute information is extracted from each paper. We define ad and id in Eq. (12) to record whether an author/institute is associated with a paper.

$$\begin{aligned} ad_{li} &= \begin{cases} 1, & \text{if } a_l \text{ is among the authors of } d_i; \\ 0, & \text{otherwise.} \end{cases} \\ id_{mi} &= \begin{cases} 1, & \text{if } i_m \text{ is among the institutes of } d_i; \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (12)$$

Second, two matrices Θ and Φ are calculated by the LDA model[4], which are the paper-topic distribution and the topic-word distribution, respectively.

Third, the relevance scores of the author a_l /institute i_m on the topic z_k , are calculated by Eq. (13) and Eq. (14), respectively.

$$p_A^{lk} = \sum_{i=1}^{N_D} ad_{li} \cdot \theta_{ik} \quad (13)$$

$$p_I^{mk} = \sum_{i=1}^{N_D} id_{mi} \cdot \theta_{ik} \quad (14)$$

Finally, as topics are anonymous and abstract concepts, we associate the topics with users' queries. The relevance score of an author a_l on a query q is derived by Eq. (15).

$$p_A^l(q) = \prod_{\{\forall j | w_j \in q\}} p_A^{lk} \quad (15)$$

where

$$\hat{k} = \arg \max_k \varphi_{kj} \quad (16)$$

Similarly, the score of an institute i_m on a query q is derived by Eq. (17).

$$p_I^m(q) = \prod_{\{\forall j | w_j \in q\}} p_I^{m\hat{k}} \quad (17)$$

With the AM and IM, we could analyze massive scientific paper resources to accurately deliver authors/institutes that are specialties at some hot research topics, as well as generate hot topic lists that are being studied by the authors/institutes, providing valuable data supports for scientific research cooperations.

4 Massive Scientific Paper Mining (MSPM) System

In this section, we will demonstrate the implementation structure of our TAIL model-based Massive Scientific Paper Mining(MSPM) System. As is shown in Fig. 1, MSPM system is set up as a B/S structure, divided into four levels: data plane, model plane, application plane and user interface.

In the data plane, MSPM system keeps crawling meta data from the Internet. At the same time, valuable information from the paper meta data is extracted, cleaned, and stored as the valid data for model-based analysis.

In the model plane, the valid data from the data plane are imported into the TAIL model, processed for the application plane. Firstly, the words are assigned to LM, LDA and CM for data preprocessing. Then AM and IM help obtain the ranking lists of authors and institutes based on the relevance scores, while the TOT model involving time stamps helps generate the topic trends. Moreover, the coauthor networks and cooperation networks among institutes could also be clearly identified.

In the application plane, various mining applications are provided based on the modeling data. Customized requirements from the user interface are delivered to the model plane, while the mining results are delivered to the user interface.

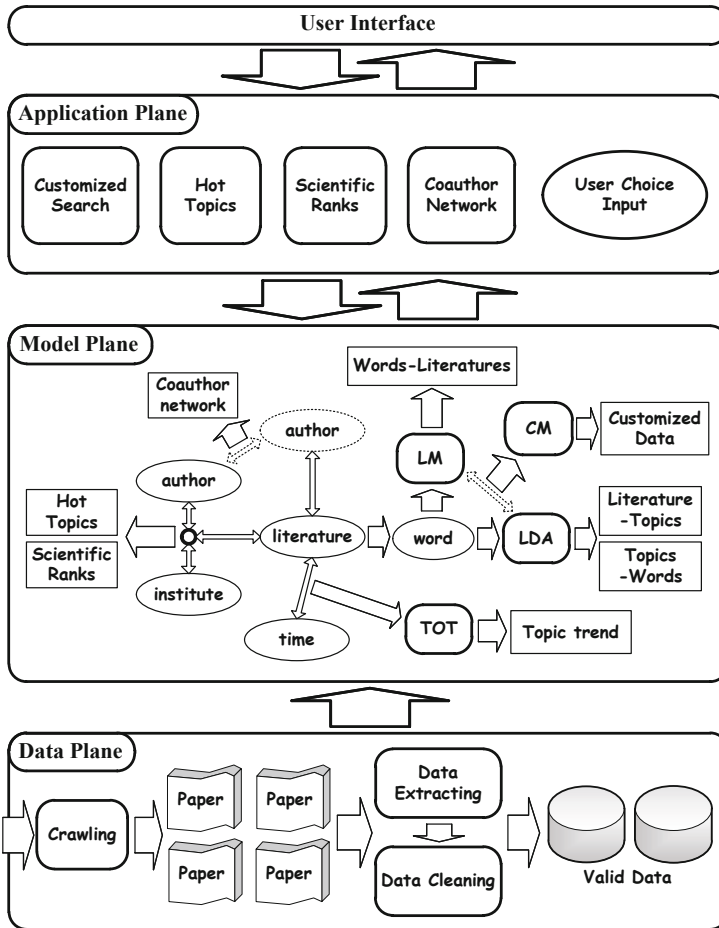


Fig. 1. System Design

5 Evaluation

To evaluate the performance of MSPM system, we experimented on large real data, containing more than 2.76 million papers from 6877 journals, published from 2005 to 2011. Each paper has structured information of title, keywords, abstract, authors, institutes, and etc.

5.1 Customized Paper Query

The basic function of MSPM system is the customized paper query. Users could provide the keywords and set the customized factor ξ , according to their query expectations. Table 2 shows the top 10 query results of the keyword “social

Table 2. Sample of Paper Query Results (keywords='social network' and $\xi = 0.2$)

Rank	Paper Title
1	Social Network Type and Subjective Well-being in a National Sample of Older Americans
2	An Experience-Sampling Study of Depressive Symptoms and Their Social Context
3	Social outcomes after temporal or extra temporal epilepsy surgery: A systematic review
4	Social dysmetria' in first-episode psychosis patients
5	Constraining heterogeneity: the social brain and its development in autism spectrum disorder
6	Viscous democracy for social networks
7	Self-concept and psychopathology in deaf adolescents: preliminary support for moderating effects of deafness-related characteristics and peer problems
8	Comparison of Anxiety-Related Traits Between Generalized and Non generalized Subtypes of Social Anxiety Disorder
9	Controllability of Boolean control networks with time delays in states
10	Minimal social network effects evident in cancer screening behavior

network” with $\xi = 0.2$, from which we can see that MSPM system could deliver closely related query results.

In addition, Fig.2 shows the trends of paper ranks with ξ changing from 0 to 1. It is clear that the ranks of some papers change dramatically with the change of ξ , while those of the other papers have no obvious changes. Thus, the customized factor could well distinguish papers of different properties.

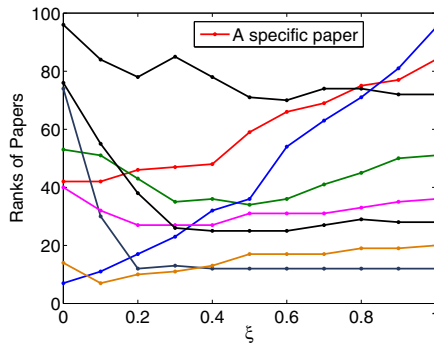


Fig. 2. Ranks of Sample Papers with Various ξ

5.2 Author/Institute Mining

With specific keyword queries, users could get the top authors and institutes with the highest relevance scores. The authors and institutes in the resulting lists are known to be the specialties in the keyword related research domains.

Table 3. Sample of Top Authors/Institutes Mining Results (keywords=‘social network’)

Top Authors	Top Institutes
Kleinberg J	The Hebrew University in Jerusalem
Jackson M	Cornell University
Wellman B	University of Maryland
Faloutsos C	Stanford University
Newman M	Carnegie Mellon University

Table 4. Precision of Different Models (**AP**=Average Precision)

Model	AP	Model	AP
LM#10	0.720	CM_Opt#5	0.852
LDA#10	0.684	CM_Opt#10	0.850
AM#10	0.768	CM_Opt#20	0.819
IM#10	0.734	CM_Opt#50	0.804

Table 3 shows the sample of top authors and institutes mining results with the keyword “social network”. Note that the results are for reference only due to data set limitations.

5.3 Precision of Query Results

To evaluate the performance of customized query, 5 graduate students are invite to judge whether the papers, authors and institutes in the returned results are relevant to their query expectations. If a paper/author/institute is rejected by more than one student, it will be regarded as irrelevant and imprecise result. We launch 50 queries with different keywords for each model, and the average precision is calculated based on the students’ judgements. As for our CM model, the query precision is further optimized with different ξ , denoted by CM_Opt.

Table 4 shows the average precisions of different models, with #N representing the precision in the top N papers. It is shown that the precision of CM is superior to that of LM [1] and LDA [4]. Moreover, CM models with larger N have lower precisions, which indicates that focusing on fewer query results will lead to better query performance. In addition, the precisions of AM and IM indicate that users could get desirable results from the author and institute mining.

6 Conclusion and Future Work

In this paper, we propose a novel TAIL model to capture the correlation of topics, time stamps, authors, institutes and literatures for massive scientific paper mining. The TAIL model defines a customized factor to balance the tradeoff of the language model and the topic model, providing customized paper queries

for users. Based on the TAIL model, we implement the Massive Scientific Paper Mining (MSPM) system and set up a B/S structure to provide web services. The evaluation results on large real data show that our MSPM system could deliver desirable mining results for various user query expectations. As for future work, our model could be further optimized by measuring paper quality and popularity based on citations, which would result in more interesting returned papers and author/institute ranking lists.

Acknowledgment. This work was supported by the National 863 Program of China (No. 2012AA011005) and Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20111102110019).

References

1. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 334–342. ACM (2001)
2. Google scholar, <http://scholar.google.com/>
3. Microsoft academic search, <http://academic.research.microsoft.com/>
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022 (2003)
5. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 306–315. ACM (2004)
6. 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)
7. Tang, J., Wang, B., Yang, Y., Hu, P., Zhao, Y., Yan, X., Gao, B., Huang, M., Xu, P., Li, W., Usadi, A.K.: Patentminer: topic-driven patent analysis and mining. In: Proceedings of the 18th ACM SIGKDD, pp. 1366–1374. ACM (2012)

Author Index

- Chang, Baobao 44
Chao, Lidia S. 280
Che, Wanxiang 52
Chen, Tongfei 25
Chen, Wei 61
Chen, Zhenbiao 61
- Dang, Jianwu 214
Du, Jinhua 291
- Fang, Qiang 214
- Gao, Guanglai 247
Ge, Tao 154
Głowińska, Katarzyna 97
Guo, Junbo 291
- Han, Dongxu 44
Hoshino, Akemi 303
Hsu, Ray 203
Hu, Junfeng 25
Hu, Minghan 257
Hu, Renfen 333
Huang, Degen 109
Huang, Xuanjing 36, 120
Huang, Zhi-e 13
- Ji, Shufan 343
Jia, Yuan 214
Jiang, Shaowei 166, 179
Jiang, Wenbin 268
Jiang, Xue 257
Jin, Jingpan 85
Jin, Yaohong 333
- Kopeć, Mateusz 97
- Lee, Tanya 203
Lei, Zhangzhang 85
Li, Aijun 214
Li, Binyang 227
Li, Fang 190
Li, Lishuang 109
Li, Ru 85
Li, Wenfeng 166, 179
- Liu, Kai 268
Liu, Song 73
Liu, Ting 52
Liu, Wuying 131
Liu, Yijia 52
Liu, Zhao 120
Lu, Yi 280
Lv, Xueqiang 25
- Min, Martin 203
Mokhtari-Fard, Iman 144
- Ogrodniczuk, Maciej 97
- Pei, Wenzhe 44
- Qiu, Xipeng 36, 120
- Rao, Gao-qi 13
Ren, Fu-Ji 73
- Savary, Agata 97
Shi, Lixin 257
Su, Jinsong 1
Su, Xiangdong 247
Sui, Zhifang 154
Sun, Maosong 238
Sun, Ruihua 315
- Tian, Le 36
- Wang, Lin 131
Wang, Longyue 280
Wang, Ning 85
Wang, Peilu 315
Wang, Sha 291
Wang, Xiaojie 166, 179
Wang, Zhiguo 325
Wang, Zhiqiang 85
Wei, Wei 61
Wei, Zhongyu 227
Wong, Derek F. 280
Wong, Kam-fai 227
- Xiao, Tong 257
Xing, Junwen 280
Xu, Bo 61

- Xu, Jun 227
Xu, Ke 343
Xue, Nianwen 325
Xuehelaiti, Miliwan 268
Xun, En-dong 13
- Yan, Xueliang 247
Yasuda, Akio 303
Yi, Mianzhu 131
Yibulayin, Tuergen 268
Yu, Dong 13
Yu, Kai 315
Yuan, Caixia 179
- Zawislawska, Magdalena 97
Zeng, Lingwei 190
Zhang, Chunliang 257
- Zhang, Kaixu 1
Zhang, Xingxing 154
Zhang, Xiyuan 291
Zhang, Yan 109
Zhao, Hai 315
Zhao, Jiayi 120
Zhao, Yu 238
Zhong, Yi-Xin 73
Zhou, Changle 1
Zhou, Huiwei 109
Zhou, Lanjun 227
Zhou, Yang 343
Zhu, Jingbo 257
Zhu, Weimeng 25
Zhu, Yun 333
Zong, Chengqing 325