IJCNLP 2008

Third International Joint Conference on Natural Language Processing

Proceedings of the Conference

Organizer

Asian Federation of Natural Language Processing

Local Host

International Institute of Information Technology, India

January 7-12 2008
Hyderabad, India

Hosts/Organizers

Asian Federation of Natural Language Processing (AFNLP)

International Institute of Information Technology, Hyderabad, India (IIIT-H)

Natural Language Processing Association of India (NLPAI)

Supporters

Yahoo! Research & Development, India

Department of Information Technology, MCIT, Government of India

Information & Communication Technology, Government of Andhra Pradesh

Microsoft Research, India

Tata Consultancy Services Limited

Centre for Development of Advanced Computing

Indian School of Business

Satyam Computer Service Limited

Google India

IBM India

Defense Research and Development Organization

Council for Scientific and Industrial Research

Preface: Conference Chair

Dear colleagues,

Welcome to the 2008 International Joint Conference on Natural Language Processing (IJCNLP-08). This is the third biennial conference organized by the Asian Federation of Natural Language Processing (AFNLP), which was founded in 2004 to promote research and development efforts in the field of computational processing of natural languages of importance to the Asian region, without regard to differences in language, race, religious belief or political stand. The first IJCNLP was held to celebrate the inauguration of AFNLP on the beautiful Hainan Island in China (March 22-24, 2004), and the second on the fantastic Jeju Island in Korea (October 10-13, 2005). Following the continuing success of the previous two conferences, the third conference is held in yet another exotic and multicultural city of Hyderabad in India in January 7-12, 2008.

On behalf of the Conference Committees, I would like to welcome all researchers and scholars who are working in all areas of Natural Language Processing (NLP) around the world and who in particular have keen interest in Asian language processing. As the world proceeds quickly into the Information Age, we face both successes and challenges in creating a global information society, and it is well recognized nowadays that Natural Language Processing provides the key to the Information Age and to solving many of these challenges, like breaking language barrier and overcoming information flood. Over the last decades, a remarkable progress has been made in NLP research and development. However, there has been a pervasive feeling that the progress of NLP for Asian languages has not been commensurate with that for Western languages. Recently the importance of Asian languages has been steadily growing as Asia becomes the dominant region of the world, economically, politically and culturally. In this context, this conference provides a forum for engineers and scientists to present and exchange their latest research findings in all aspects of NLP and thus to promote research and development activities for Asian language processing. This is the major motivation of IJCNLP.

I would like to express my sincere appreciation to the authors of invited and contributed papers and to all conference participants for their active participations. I also wish to express my heartfelt gratitude and thanks to the Committee Members, particularly the Organizing Co-chairs Rajeev Sangal and Raji Bagga, the Program Co-chairs Yuji Matsumoto and Ann Copestake, the Publication Chair Jing-Shin Chang, and all the other Committee Chairs for their tremendous efforts and substantial contributions to the conference. I feel honored and blessed to be part of this conference as the Conference Chair working with such wonderful team. With our team efforts, I am confident that this conference will be even more successful than the previous. Finally, I hope that you will participate actively in all sessions and events to maximize the benefits from them, and I also wish all participants a very fruitful and enjoyable time during the conference in Hyderabad.

Jong-Hyeok Lee
Conference Chair

Preface: Program Committee Co-Chairs

This volume contains the papers accepted for presentation at the third International Joint Conference on Natural Language Processing (IJCNLP-2008). IJCNLP is held approximately every two years as the flagship conference of the AFNLP (Asian Federation of Natural Language Processing). This year's conference, which follows the success of IJCNLP-2005 on Jeju Island in Korea, is in the city which is such a beautiful mixture of ancient civilization and modern industry: Hyderabad, India.

On behalf of the Program Committee, we are pleased to present this volume, which includes the accepted papers for oral and poster presentations at the conference. We received 266 submissions from 28 different regions all over the world; 74% from Asia, 15% from North America, 9% from Europe, 3% from Australia, and 0.4% from Africa.

The paper selection was not easy with this large number of submission but with the devoted work of our 13 area chairs and 268 PC members, we were able to select very high quality papers. 75 papers (27.8%) were accepted for oral presentation and 62 papers (23.3%) were accepted for poster presentation. After a few withdrawals, this volume contains 74 oral papers and 60 poster papers.

We are also very grateful to Professor Aravind Joshi (University of Pennsylvania), Professor Hyopil Shin (Seoul National University) and Dr S.H. Srinivasan (Yahoo!) for accepting to give keynote and invited talks, which surely make the conference more attractive.

Organizing and hosting this size of international conference requires lots of help and effort from many people. We would like to send our greatest thanks to the Honorary Conference Chair, Professor Aravind Joshi and the Conference Chair, Professor Jong-Hyoek Lee for their continuous support and timely guidance. We would also thank the Local Organizing Co-Chairs Professor Rajeev Sangal and Dr Raji Bagga for their support, advice and responses to our numerous requests. Special thanks are due to the Publication Chair Professor Jing-Shin Chang. Without his very detailed format checking and efficient compilation, this volume of proceedings would not come out in the current form. We would like to express our highest gratitude to all the other Committee Chairs. Working with such a wonderful group of people has been great fun. Last but not least, we would like to thank all the people who submitted their papers and all the people who attend this conference.

Welcome to IJCNLP-2008. We hope you enjoy this conference as much as we do.

Ann Copestake and Yuji Matsumoto
Program Committee Co-Chairs

Keynote Speech:

PENN Discourse Treebank: Complexity of Dependencies at the Discourse Level and at the Sentence Level

Aravind K. Joshi
Department of Computer and Information Science and
Institute for Research in Cognitive Science
University of Pennsylvania, Philadelphia, PA, USA

ABSTRACT

First, I will describe the Penn Discourse Treebank (PDTB)*, a corpus in which we annotate the discourse connectives (explicit and implicit) and their arguments, together with "attributions" of the arguments and the relations denoted by the connectives, and also the senses of the connectives. I will then discuss some issues concerning the complexity of dependencies in terms of the elements that bear the dependency relations, the graph theoretic properties of these dependencies such as nested and crossed dependencies, dependencies with shared arguments, and finally, the attributions and their relationship to the dependencies, among others. We will compare these dependencies with those at the sentence level and then discuss some aspects that relate to the transition from the sentence level to the level of "immediate discourse" and propose some conjectures.

---

* This 1 million-word corpus is the same as the WSJ corpus used by the Penn Treebank (PTB) for syntactic annotation and by Propbank for predicate-argument annotation. PDTB 2.0 will be released by the Linguistic Data Consortium (LDC) in early February 2008.

Members of the PDTB project: Nikhil Dinesh, Aravind K. Joshi, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, and Bonnie Webber (University of Edinburgh).

Invited Talk:

The 21st Sejong Project: with a Focus on Building of the SELK (Sejong Electronic Lexicon of Korean) and the KNC (Korean National Corpus)

Hyopil Shin
Dept. of Linguistics, Seoul National University
School of Computer Engineering, Seoul National University

ABSTRACT

The 21st Sejong Project started in 1998 with a 10-year plan. The project was funded by the Ministry of Culture and Tourism of the Korean government. The goal of the project was to promote technological expertise in Korean language research and technology. The project consists of 8 sub-projects ranging from construction of Korean language resources to management and distribution of outputs from the work. The core part of the project is to compile an electronic lexical dictionary and to build a large-scale Korean corpus.

The SELK focuses on an exhaustive representation of Korean linguistic knowledge by harmonizing linguistic validity, psychological reality, and computational efficiency. The SELK is composed of various sub-dictionaries corresponding to the parts-of-speech-based word categories such as nouns, verbs, adverbs etc. The lexicon shows a considerable differentiation from other paperback or machine-readable dictionaries in Korean in its precise and comprehensive representation.

The KNC project has two sub-divisions, one for a general corpus and the other for a special corpus. The general corpus division collected a wide range of unconstrained materials and endeavored at annotating the data with parts-of-speech, syntactic, and semantic tags. The special data division, on the other hand, constructed Korean-English and Korean-Japanese corpora, a historical corpus, and a corpus used by North Koreans and overseas Koreans.

The SELK and the KNC are beginning to serve as important research tools for investigators in natural language processing as well as in theoretical linguistics. Annotated corpora and well-established electronic dictionaries promise to be valuable for enterprises such as the construction of statistical models for the grammar of written and spoken Korean, the development of software for Korean language processing, and even the publication of the paperback Korean dictionaries.

In this speech, I will introduce the 21st Sejong Project and review my experience with constructing one such large language resource - the SELK, consisting of about 600,000 lexical entries, and the KNC, consisting of about 500 million word collections. Considering the size and time needed to develop it, this project deserves great attention. We, however, also experienced a lot of difficulties through trial and error, inevitably originating from such a long work period and the large scale of the work. We hope sharing such experiences will help researchers with the same interests, to break through the obstacles and to avoid mistakes we have made for a decade.

Invited Talk:

Language Processing for the Evolving Web

Srinivasan Sengamedu
Yahoo!, Bangalore

ABSTRACT

World Wide Web brings several new dimensions to language processing - social, multimodal, structural, etc. The social dimension arises from the tagging phenomenon, multimodal from the coexistence of images and videos with text in web documents, and structure from rich formatting of web pages. While the massive amounts of data available has made new approaches to translation, summarization, and extraction possible, next generation applications like semantic search require radically new theoretical ideas. The talk will outline the phenomena, summarize recent achievements, and pose the new challenges.

Conference Committee

Honorary Conference Chair

Prof. Aravind Joshi, University of Pennsylvania

Conference Chair

Prof. Jong-Hyeok Lee, POSTECH, Korea

Local Organizing Co-chairs

Prof. Rajeev Sangal, International Institute of Information Technology, India
Dr. Raji Bagga, International Institute of Information Technology, India

LOC Steering/Core Committee Members

Dr. A Kumaran, Microsoft Research India
Dr. Pushpak Bhattacharya, IIT Bombay, India
Dr. M Sasikumar, CDAC, Mumbai, India
Dr. Dipti Misra Sharma, International Institute of Information Technology, India
Dr. Vasudeva Varma, International Institute of Information Technology, India
Dr. KS Rajan, International Institute of Information Technology, India
Dr. Anoop M Namboodri, International Institute of Information Technology, India
Mr. KS Vijaya Sekhar, IJCNLP-08 LOC Secretariat

Program Committee Co-chairs

Prof. Yuji Matsumoto, NAIST, Japan
Prof. Ann Copestake, University of Cambridge, UK

Publication Committee Chair

Prof. Jing-Shin Chang, National Chi-Nan University, Taiwan

Publicity Committee Co-chairs

Prof. Satoshi Sato, Nagoya University, Japan
Prof. Jian-Yun Nie, Univ. of Montreal, Canada

Financial Committee Co-chairs

Prof. Kam-Fai Wong, Chinese Univ. of HK, Hong Kong
Dr. A. Kumaran, Microsoft Research India

Tutorial Committee Co-chairs

Prof. Kemal Oflazer, Sabanci University, Turkey
Prof. Key-Sun Choi, KAIST, Korea

Workshops Committee Co-chairs

Dr. Haizhou Li, Institute of Infocomm Research, Singapore
Dr. Timothy Baldwin, University of Melbourne, Australia

Exhibition/Demo Committee Co-chairs

Prof. Pushpak Bhattacharya, Indian Institute of Technology Bombay, India
Prof. Fei Xia, Univ. of Washington, USA

Program Committee

Program Committee Chairs:

Ann Copestake, University of Cambridge
(Ann.Copestake@cl.cam.ac.uk)
Yuji Matsumoto, Nara Institute of Science and Technology
(matsu@is.naist.jp)

Areas and Area Chairs:

Information Retrieval
Hang Li, Microsoft China

Parsing/Grammatical Formalisms
Beth Ann Hockey, UCSC

Chunking/Shallow Parsing
Srinivas Bangalore, AT&T Research

Statistical Models/Machine Learning for NLP
Robert Malouf, San Diego State University

Machine Translation
Philipp Koehn, University of Edinburgh

Word Segmentation/POS Tagging
Yuji Matsumoto, Nara Institute of Science and Technology

Semantic Processing/Lexical Semantics
Patrick St Dizier, IRIT

Ontologies and Linguistic Resources
Nancy Ide, Vassar College

Paraphrasing/Entailment/Generation
Kentaro Inui, NAIST

Discourse
Alistair Knott, University of Otago

QA/Text Summarization
Sanda Harabagiu, University of Texas at Dallas

Text Mining/Information Extraction
James Curran, University of Sydney

Spoken Language Processing
Gary Geunbae Lee, POSTECH

Asian Language Processing and Linguistics Issues in NLP
Chu-Ren Huang, Academia Sinica

Program Committee Members:

Eugene Agichtein, Yaser Al-Onaizan, James Allen, Pascal Amsili, Alina Andreevskaia, Kenji Araki, Nick Asher

Collin Baker, Jason Baldridge, Timothy Baldwin, Sivaji Bandyopadhyay, Srinivas angalore, Marco Baroni, Stephen Beale, Tilman Becker, Susham Bendre, Pushpak Bhattacharyya, Zhao Bing, Sasha Blair-Goldensohn, Francis Bond, Kalina ontcheva, Johan Bos, Pierrette Bouillon, Antonio Branco, Thorsten Brants, Paul Buitelaar, Razvan Bunescu, Harry Bunt, Miriam Butt, Ekaterina Buyko, Bill Byrne

Aoife Cahill, Chris Callison-Burch, Nicoletta Calzolari, Joyce Chai, Hsin-Hsi Chen, Keh-Jiann Chen, David Chiang, Massimiliano Ciaramita, Philipp Cimiano, tephen Clark, Simon Clematide, Trevor Cohn, Nigel Collier, John Conroy, Nick Craswell, Dan Cristea, Andras Csomai, Silviu Cucerzan, Hang Cui

Walter Daelemans, Robert Dale, Hal Daume, Thierry Declerck, Fernando Diaz, Hakkani-Tur Dilek, Pavel Dmitriev, Bill Dolan, Christy Doran, Bonnie Dorr, Mark Dras, Markus Dreyer

Cecile Fabre, Hui Fang, Marcello Federico, Christiane Fellbaum, Dan Flickinger, J.Foster, Alex Fraser, Atsushi Fujita

Bin Gao, Claire Gardent, Albert Gatt, Tanja Gaustad, Niyu Ge, Dan Gildea, Jade Goldstein-Stewart, Ralph Grishman, Claire Grover, Narendra Gupta

Patrick Haffner, Jan Hajic, Keith Hall, Sanda Harabagiu, Samer Hassan, Toby Hawker, Mary Hearne, John Henderson, Andrew Hickl, Deidre Hogan, Tracy Holloway King, Shu-Kai Hsieh, Sarmad Hussain

Kentaro Inui, Hitoshi Isahara, Masato Ishizaki

Donghong Ji

Laura Kallmeyer, Kyoko Kanzaki, Tatsuya Kawahara, Asanee Kawtrakul, Junichi Kazama, Andrew Kehler, Bernd Kiefer, Adam Killgariff, Byeongchang Kim, Jin Dong Kim, Atanis Kiryakov, Manfred Klenner, Kevin Knight, Anna Korhonen, Emiel Krahmer, Shankar Kumar, Sadoa Kurohashi

Philippe Langlais, Mirella Lapata, Alon Lavie, Kiyong Lee, Alessandro Lenci, Yves Lepage, Baoli Li, Haizhou Li, Wenjie Li, Chin-Yew Lin, Tie-Yan Liu, Adam Lopez

Ryan MacDonald, Bernardo Magnini, Steve Maiorano, Robert Malouf, Gideon Mann,Alda Mari, Katja Markert, Carlos Martin-Vide, Kathleen McCoy, Ryan McDonald, Marge McShane, Michael McTear, Arul Menezes, Helen Meng, Wolfgang Minker,Vibhu Mittal, Yusuke Miyao, Paola Monachesi, Bob Moore, Tony Mullen

Program Committee Members (cont.) :

Sobha Nair, Hiroshi Nakagawa, Seiichi Nakagawa, Satoshi Nakamura, Shri Narayanan, Srini Narayanan, Vivi Nastase, Tetsuya Nasukawa, Roberto Navigli, Vincent Ng, Orace Ngai, Thi Minh Huyen Nguyen, Jianyun Nie, Joakim Nivre, Yongkyoon No, Tadashi Nomoto, Eric Nyberg

Franz Och, Stephen Oepen, Kemal Oflazer, Miles Osborne, Lilja Ovreli

Martha Palmer, Shimei Pan, Antonio Pareja-Lora, Jong Park, Jon Patrick, Wim Peters, Manfred Pinkal, Paul Piwek, Thierry Poibeau, David Powers, Kishore Prahallad, Rashmi Prasad

Chris Quirk

Allan Ramsay, Lance Ramshaw, Manny Rayner, Christian Retore, Brian Roark, Horacio Rodri'guez, Antti-Veikko Rosti, Rachel E. O. Roxas, Thomas Russ

Kenji Sagae, Horacio Saggion, Patrick Saint-Dizier, Ted Sanders, Rajeev Sangal, Anoop Sarkar, Sudeshna Sarkar, Holger Schwenk, Donia Scott, Satoshi Sekine, Burr Settles, Vijay Shanker, Dipiti Misra Sharma, Libin Shen, Kiyoaki Shirai, Candy Sidner, Khalil Simaan, Michel Simard, Navjyoti Singh, Noah A Smith, David Smith, Virach Sornlertlamvanich, Wilbert Spooren, Caroline Sporleder, Manfred Stede, Mark Stevenson, Kristina Striegnitz, Michael Strube, Craig Struble, Jian Su, Mihai Sudreanu, Eiichiro Sumita, Yoshimi Suzuki, Hisami Suzuki

Jie Tang, Simone Teufel, Thanaruk Theeramunkong, Jo"rg Tiedemann, Christoph Tillmann, Ivan Titov, Takenobu Tokunaga, Shisanu Tongchim, Kentaro Torisawa, Kristina Toutanova, David Traum, Shu-Chuan Tseng, Yoshimasa Tsuruoka

Nicola Ueffing

Ielka van der Sluis, Menno van Zaanen, Lucy Vanderwende, Vasudeva Varma, Sriram Venkatapathy, Ashish Venugopal, Jette Viethen, Andreas Vlachos, Stephan Vogel

Marilyn Walker, Hsin-min Wang, Taro Watanabe, Bonnie Webber, Ralph Weischedel, Casey Whitelaw, Yuk Wah Wong, Yunfang Wu, Dekai Wu

Peng Xu, Jun Xu, Nianwen Xue

Naoki Yoshinaga, Yong Yu

Annie Zaenen, Richard Zens, Jun Zhao, Ming Zhou, GuoDong Zhou, Jing-bo Zhu, Michael Zock

Table of Contents

Volume I

Ontologies and Linguistic Resources (3)

Semantic Similarity

Volume II

Poster papers (Poster session 1)

Tutorial

Demo

A Lemmatization Method for Modern Mongolian and its Application to Information Retrieval

Badam-Osor Khaltar **Atsushi Fujii**
Graduate School of Library, Information and Media Studies
University of Tsukuba
1-2 Kasuga Tsukuba, 305-8550, Japan
{khab23, fujii}@slis.tsukuba.ac.jp

Abstract

In Modern Mongolian, a content word can be inflected when concatenated with suffixes. Identifying the original forms of content words is crucial for natural language processing and information retrieval. We propose a lemmatization method for Modern Mongolian and apply our method to indexing for information retrieval. We use technical abstracts to show the effectiveness of our method experimentally.

1 Introduction

The Mongolian language is divided into Traditional Mongolian, which uses the Mongolian alphabet, and Modern Mongolian, which uses the Cyrillic alphabet. In this paper, we focus solely on the latter and use the word "Mongolian" to refer to Modern Mongolian.

In Mongolian, which is an agglutinative language, each sentence is segmented on a phrase-by-phrase basis. A phrase consists of a content word, such as a noun or a verb, and one or more suffixes, such as postpositional participles. A content word can potentially be inflected when concatenated with suffixes.

Identifying the original forms of content words in Mongolian text is crucial for natural language processing and information retrieval. In information retrieval, the process of normalizing index terms is important, and can be divided into lemmatization and stemming. Lemmatization identifies the original form of an inflected word, whereas stemming identifies a stem, which is not necessarily a word.

Existing search engines, such as Google and Yahoo!, do not perform lemmatization or stemming for indexing Web pages in Mongolian. Therefore, Web pages that include only inflected forms of a query cannot be retrieved.

In this paper, we propose a lemmatization method for Mongolian and apply our method to indexing for information retrieval.

2 Inflection types in Mongolian phrases

Nouns, adjectives, numerals, and verbs can be concatenated with suffixes. Nouns and adjectives are usually concatenated with a sequence of a plural suffix, case suffix, and reflexive possessive suffix. Numerals are concatenated with either a case suffix or a reflexive possessive suffix. Verbs are concatenated with various suffixes, such as an aspect suffix, a participle suffix, and a mood suffix.

Figure 1 shows the inflection types of content words in Mongolian phrases. In (a), there is no inflection in the content word "ном (book)", concatenated with the suffix "ын (the genitive case)". The content words are inflected in (b)-(e).

Type	Example
(a) No inflection	**ном + ын → номын** book + genitive case
(b) Vowel insertion	**ах + д → ахад** brother + dative case
(c) Consonant insertion	**байшин + ийн→ байшингийн** building + genitive case
(d) The letters "ь" or "и" are eliminated, and the vowel converts to "и"	**анги + аас → ангиас** return + ablative case
(e) Vowel elimination	**ажил + аас → ажлаас** work + ablative case

Figure 1: Inflection types of content words in Mongolian phrases.

Loanwords, which can be nouns, adjectives, or verbs in Mongolian, can also be concatenated with suffixes. In this paper, we define a loanword as a word imported from a Western language.

Because loanwords are linguistically different from conventional Mongolian words, the suffix concatenation is also different from that for conventional Mongolian words. Thus, exception rules are required for loanwords.

For example, if the loanword "**станц** (station)" is to be concatenated with a genitive case suffix, "**ын**" should be selected from the five genitive case suffixes (i.e., **ын**, **ийн**, **ы**, **ий**, and **н**) based on the Mongolian grammar. However, because "**станц** (station)" is a loanword, the genitive case "**ийн**" is selected instead of "**ын**", resulting in the noun phrase "**станцийн** (station's)".

Additionally, the inflection (e) in Figure 1 never occurs for noun and adjective loanwords.

3 Related work

Sanduijav et al. (2005) proposed a lemmatization method for noun and verb phrases in Mongolian. They manually produced inflection rules and concatenation rules for nouns and verbs. Then, they automatically produced a dictionary by aligning nouns or verbs with suffixes. Lemmatization for phrases is performed by consulting this dictionary.

Ehara et al. (2004) proposed a morphological analysis method for Mongolian, for which they manually produced rules for inflections and concatenations. However, because the lemmatization methods proposed by Sanduijav et al. (2005) and Ehara et al. (2004) rely on dictionaries, these methods cannot lemmatize new words that are not in dictionaries, such as loanwords and technical terms.

Khaltar et al. (2006) proposed a lemmatization method for Mongolian noun phrases that does not use a noun dictionary. Their method can be used for nouns, adjectives, and numerals, because the suffixes that are concatenated with these are almost the same and the inflection types are also the same. However, they were not aware of the applicability of their method to adjectives and numerals.

The method proposed by Khaltar et al. (2006) mistakenly extracts loanwords with endings that are different from conventional Mongolian words. For example, if the phrase "**экологийн** (ecology's)" is lemmatized, the resulting content word will be "**эколог**", which is incorrect. The correct word is "**экологи** (ecology)". This error occurs because the ending "**-ологи** (-ology)" does not appear in conventional Mongolian words.

In addition, Khaltar et al. (2006)'s method applies (e) in Figure 1 to loanwords, whereas inflection (e) never occurs in noun and adjective loanwords.

Lemmatization and stemming are arguably effective for indexing in information retrieval (Hull, 1996; Porter, 1980). Stemmers have been developed for a number of agglutinative languages, including Malay (Tai et al., 2000), Indonesian (Berlian Vega and Bressan, 2001), Finnish (Korenius et al., 2004), Arabic (Larkey et al., 2002), Swedish (Carlberger et al., 2001), Slovene (Popovič and Willett, 1992) and Turkish (Ekmekçioglu et al., 1996).

Xu and Croft (1998) and Melucci and Orio (2003) independently proposed a language-independent method for stemming, which analyzes a corpus in a target language and identifies an equivalent class consisting of an original form, inflected forms, and derivations. However, their method, which cannot identify the original form in each class, cannot be used for natural language applications where word occurrences must be standardized by their original forms.

Finite State Transducers (FSTs) have been applied to lemmatization. Although Karttunen and Beesley (2003) suggested the applicability of FSTs to various languages, no rule has actually been proposed for Mongolian. The rules proposed in this paper can potentially be used for FSTs.

To the best of our knowledge, no attempt has been made to apply lemmatization or stemming to information retrieval for Mongolian. Our research is the first serious effort to address this problem.

4 Methodology

4.1 Overview

In view of the discussion in Section 3, we enhanced the lemmatization method proposed by Khaltar et al. (2006). The strength of this method is that noun dictionaries are not required.

Figure 2 shows the overview of our lemmatization method for Mongolian. Our method consists of two segments, which are identified with dashed lines in Figure 2: "lemmatization for verb phrases" and "lemmatization for noun phrases".

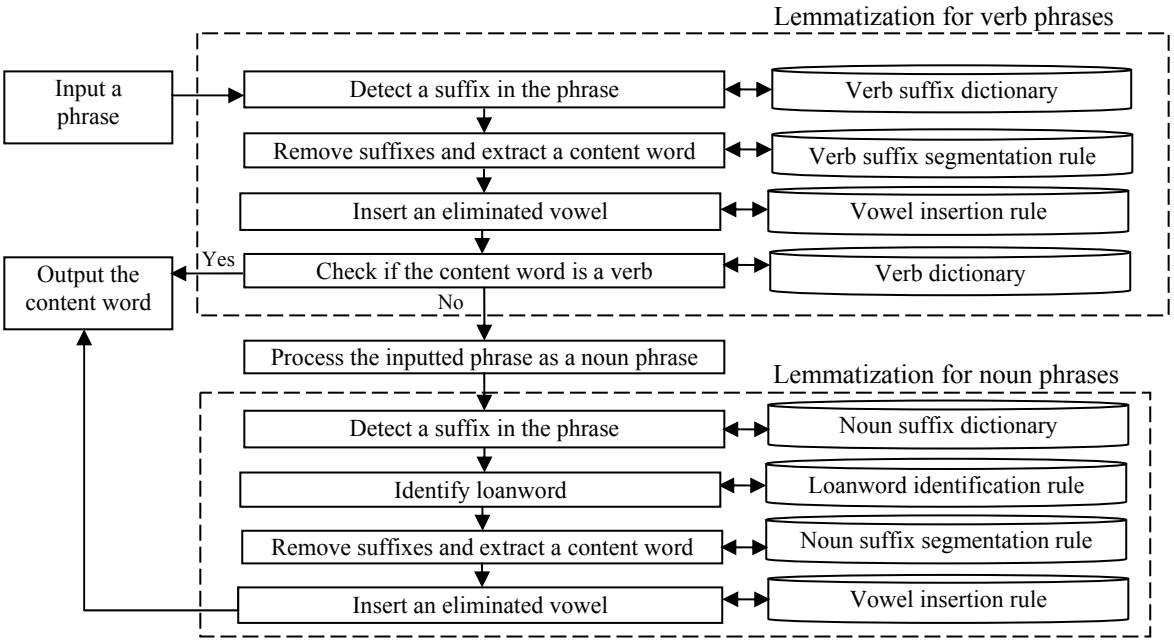


Figure 2: Overview of our lemmatization method for Mongolian.

In Figure 2, we enhanced the method proposed by Khaltar et al. (2006) from three perspectives.

First, we introduced "lemmatization for verb phrases". There is a problem to be solved when we target both noun and verb phrases. There are a number of suffixes that can concatenate with both verbs and nouns, but the inflection type can be different depending on the part of speech. As a result, verb phrases can incorrectly be lemmatized as noun phrases and vice versa.

Because new verbs are not created as frequently as nouns, we predefine a verb dictionary, but do not use a noun dictionary. We first lemmatize an entered phrase as a verb phrase and then check whether the extracted content word is defined in our verb dictionary. If the content word is not defined in our verb dictionary, we lemmatize the input phrase as a noun phrase.

Second, we introduced a "loanword identification rule" in "lemmatization for noun phrases". We identify a loanword phrase before applying a "noun suffix segmentation rule" and "vowel insertion rule". Because segmentation rules are different for conventional Mongolian words and loanwords, we enhance the noun suffix segmentation rule that was originally proposed by Khaltar et al. (2006). Additionally, we do not use the vowel insertion rule, if the entered phrase is detected as a loanword phrase. The reason is that vowel elimination never occurs in noun loanwords.

Third, unlike Khaltar et al. (2006), we targeted adjective and numeral phrases. Because the suffixes concatenated with nouns, adjectives, and numerals are almost the same, the lemmatization method for noun phrases can also be used for adjective and numeral phrases without any modifications. We use "lemmatization for noun phrases" to refer to the lemmatization for noun, adjective, and numeral phrases.

We briefly explain our lemmatization process using Figure 2.

We consult a "verb suffix dictionary" and perform backward partial matching to determine whether a suffix is concatenated at the end of a phrase. If a suffix is detected, we use a "verb suffix segmentation rule" to remove the suffix and extract the content word. This process will be repeated until the residue of the phrase does not match any of the entries in the verb suffix dictionary.

We use a "vowel insertion rule" to check whether vowel elimination occurred in the content word and insert the eliminated vowel.

If the content word is defined in a "verb dictionary", we output the content word as a verb and terminate the lemmatization process. If not, we use the entered phrase and perform lemmatization for noun phrases. We consult a "noun suffix dictionary" to determine whether one or more suffixes are concatenated at the end of the target phrase.

We use a "loanword identification rule" to identify whether the phrase is a loanword phrase. We use a "noun suffix segmentation rule" to remove the suffixes and extract the content word. If the phrase is identified as a loanword phrase we use different segmentation rules.

We use the "vowel insertion rule" which is also used for verb phrases to check whether vowel elimination occurred in the content word and insert the eliminated vowel. However, if the phrase is identified as a loanword phrase, we do not use the vowel insertion rule.

If the target phrase does not match any of the entries in the noun suffix dictionary, we determine that a suffix is not concatenated and we output the phrase as it is.

The inflection types (b)–(d) in Figure 1 are processed by the verb suffix segmentation rule and noun suffix segmentation rule. The inflection (e) in Figure 1 is processed by the vowel insertion rule.

We elaborate on the dictionaries and rules in Sections 4.2–4.8.

4.2 Verb suffix dictionary

We produced a verb suffix dictionary, which consists of 126 suffixes that can concatenate with verbs. These suffixes include aspect suffixes, participle suffixes, and mood suffixes.

Figure 3 shows a fragment of our verb suffix dictionary, in which inflected forms of suffixes are shown in parentheses. All suffixes corresponding to the same suffix type represent the same meaning.

4.3 Verb suffix segmentation rule

For the verb suffix segmentation rule, we produced 179 rules. There are one or more segmentation rules for each of the 126 verb suffixes mentioned in Section 4.2.

Figure 4 shows a fragment of the verb suffix segmentation rule for suffix "**в** (past)". In the column "Segmentation rule", the condition of each "if" sentence is a phrase ending. "V" refers to a vowel and "*" refers to any strings. "C9" refers to any of the nine consonants "**ц**", "**ж**", "**з**", "**с**", "**д**", "**т**", "**ш**", "**ч**", or "**х**", and "C7" refers to any of the seven consonants "**м**", "**г**", "**н**", "**л**", "**б**", "**в**", or "**р**". If a condition is satisfied, we remove one or more corresponding characters.

For example, because the verb phrase "**шинэчлэв** (renew + past)" satisfies condition (ii), we remove the suffix "**в**" and the preceding vowel "**э**" to extract "**шинэчл**".

Suffix type	Suffix
Appeal	**гтун, гтүн**
Complete	**чих**
Perfect	**аад (иад), оод (иод), ээд, өөд**
Progressive-perfect	**саар, соор, сээр, сөөр**

Figure 3: Fragment of verb suffix dictionary.

Suffix	Segmentation rule
в Past	(i) If (*+ V + V + **в**) Remove **в**
	(ii) If (* + C9 + C7 + V + **в**) Remove V + **в**

Figure 4: Fragment of verb suffix segmentation rule.

4.4 Verb dictionary

We use the verb dictionary produced by Sanduijav et al. (2005), which includes 1254 verbs.

4.5 Noun suffix dictionary

We use the noun suffix dictionary produced by Khaltar et al. (2006), which contains 35 suffixes that can be concatenated with nouns. These suffixes are postpositional particles. Figure 5 shows a fragment of the dictionary, in which inflected forms of suffixes are shown in parentheses.

4.6 Noun suffix segmentation rule

There are 196 noun suffix segmentation rules, of which 173 were proposed by Khaltar et al. (2006). As we explained in Section 3, these 173 rules often incorrectly lemmatize loanwords with different endings from conventional Mongolian words.

We analyzed the list of English suffixes and found that English suffixes "-ation" and "-ology" are incorrectly lemmatized by Khaltar et al. (2006). In Mongolian, "-ation" is transliterated into "**ации**" or "**яци**" and "-ology" is transliterated into "**ологи**". Thus, we produced 23 rules for loanwords that end with "**ации**", "**яци**", or "**ологи**".

Figure 6 shows a fragment of our suffix segmentation rule for loanwords. For example, for the loanword phrase "**экологийн** (ecology + genitive)", we use the segmentation rule for suffix "**ийн** (genitive)" in Figure 6. We remove the suffix "**ийн** (genitive)" and add "**и**" to the end of the content word. As a result, the noun "**экологи** (ecology)" is correctly extracted.

Case	Suffix
Genitive	**н, ы, ын, ий, ийн**
Accusative	**ыг, ийг, г**
Dative	**д, т**
Ablative	**аас (иас), оос (иос), ээс, өөс**

Figure 5: Fragment of noun suffix dictionary.

Suffix	Segmentation rule for loanwords
ийн Genitive	If (* + **логийн**) Remove (**ийн**) , Add (**и**)
ийг Accusative	If (* + **логийг**) Remove (**ийг**), Add (**и**)

Figure 6: Fragment of suffix segmentation rules for loanwords.

4.7 Vowel insertion rule

To insert an eliminated vowel and extract the original form of a content word, we check the last two characters of the content word. If they are both consonants, we determine that a vowel was eliminated. However, a number of Mongolian words end with two consonants inherently and, therefore, Khaltar et al. (2006) referred to a textbook on the Mongolian grammar (Ts, 2002) to produce 12 rules to determine when to insert a vowel between two consecutive consonants. We also use these rules as our vowel insertion rule.

4.8 Loanword identification rule

Khaltar et al. (2006) proposed rules for extracting loanwords from Mongolian corpora. Words that satisfy one of seven conditions are extracted as loanwords. Of the seven conditions, we do not use the condition that extracts a word ending with "consonants + **и**" as a loanword because it was not effective for lemmatization purposes in preliminary study.

5 Experiments

5.1 Evaluation method

We collected 1102 technical abstracts from the "Mongolian IT Park" [1] and used them for experiments. There were 178,448 phrase tokens and 17,709 phrase types in the 1102 technical abstracts. We evaluated the accuracy of our lemmatization method (Section 5.2) and the effectiveness of our method in information retrieval (Section 5.3) experimentally.

[1] http://www.itpark.mn/ (October, 2007)

5.2 Evaluating lemmatization

Two Mongolian graduate students served as assessors. Neither of the assessors was an author of this paper. The assessors provided the correct answers for lemmatization. The assessors also tagged each word with its part of speech.

The two assessors performed the same task independently. Differences can occur between two assessors on this task. We measured the agreement of the two assessors by the Kappa coefficient, which ranges from 0 to 1. The Kappa coefficients for performing lemmatization and tagging of parts of speech were 0.96 and 0.94, respectively, which represents almost perfect agreement (Landis and Koch, 1977). However, to enhance the objectivity of the evaluation, we used only the phrases for which the two assessors agreed with respect to the part of speech and lemmatization.

We were able to use the noun and verb dictionaries of Sanduijav et al. (2005). Therefore, we compared our lemmatization method with Sanduijav et al. (2005) and Khaltar et al. (2006) in terms of accuracy.

Accuracy is the ratio of the number of phrases correctly lemmatized by the method under evaluation to the total number of target phrases. Here, the target phrases are noun, verb, adjective, and numeral phrases.

Table 1 shows the results of lemmatization. We targeted 15,478 phrase types in the technical abstracts. Our experiment is the largest evaluation for Mongolian lemmatization in the literature. In contrast, Sanduijav et al. (2005) and Khaltar et al. (2006) used only 680 and 1167 phrase types, respectively, for evaluation purposes.

In Table 1, the accuracy of our method for nouns, which were targeted in all three methods, was higher than those of Sanduijav et al. (2005) and Khaltar et al. (2006). Because our method and that of Sanduijav et al. (2005) used the same verb dictionary, the accuracy for verbs is principally the same for both methods. The accuracy for verbs was low, because a number of verbs were not included in the verb dictionary and were mistakenly lemmatized as noun phrases. However, this problem will be solved by enhancing the verb dictionary in the future. In total, the accuracy of our method was higher than those of Sanduijav et al. (2005) and Khaltar et al. (2006).

Table 1: Accuracy of lemmatization (%).

	#Phrase types	Sanduijav et al. (2005)	Khaltar et al. (2006)	Our method
Noun	13,016	57.6	87.7	92.5
Verb	1,797	24.5	23.8	24.5
Adjective	609	82.6	83.5	83.9
Numeral	56	41.1	80.4	81.2
Total	15,478	63.2	72.3	78.2

We analyzed the errors caused by our method in Figure 7. In the column "Example", the left side and the right side of an arrow denote an error and the correct answer, respectively.

The error (a) occurred to nouns, adjectives, and numerals, in which the ending of a content word was mistakenly recognized as a suffix and was removed. The error (b) occurred because we did not consider irregular nouns. The error (c) occurred to loanword nouns because the loanword identification rule was not sufficient. The error (d) occurred because we relied on a verb dictionary. The error (e) occurred because a number of nouns were incorrectly lemmatized as verbs.

For the errors (a)-(c), we have not found solutions. The error (d) can be solved by enhancing the verb dictionary in the future. If we are able to use part of speech information, we can solve the error (e). There are a number of automatic methods for tagging parts of speech (Brill, 1997), which have promise for alleviating the error (e).

5.3 Evaluating the effectiveness of lemmatization in information retrieval

We evaluated the effectiveness of lemmatization methods in indexing for information retrieval. No test collection for Mongolian information retrieval is available to the public. We used the 1102 technical abstracts to produce our test collection.

Figure 8 shows an example technical abstract, in which the title is "Advanced Albumin Fusion Technology" in English. Each technical abstract contains one or more keywords. In Figure 8, keywords, such as "**цусны ийлдэс** (blood serum)" and "**эхэс** (placenta)" are annotated.

We used two different types of queries for our evaluation. First, we used each keyword as a query, which we call "keyword query (KQ)". Second, we used each keyword list as a query, which we call "list query (LQ)". The average number for keywords in the keywords list was 6.1. For each query,

Reasons of errors	#Errors	Example
(a) Word ending is the same as a suffix.	274	**сорт → сор** sort
(b) Noun plural tense is irregular.	244	**амьтан → амьт** animal
(c) Noun loanword ends with two consonants.	94	**динозавр → динозавар** dinosaur
(d) Verb does not exist in our verb dictionary.	689	**кодло → кодлох** to code
(e) Word corresponds to multiple part of speech.	853	**орон → ор** country inter

Figure 7: Errors of our lemmatization method.

we used as the relevant documents the abstracts that were annotated with the query keyword in the keywords field. Thus, we were able to avoid the cost of relevance judgments.

The target documents are the 1102 technical abstracts, from which we extracted content words in the title, abstract, and result fields as index terms. However, we did not use the keywords field for indexing purposes. We used Okapi BM25 (Robertson et al., 1995) as the retrieval model.

We used the lemmatization methods in Table 2 to extract content words and compared the Mean Average Precision (MAP) of each method using KQ and LQ. MAP has commonly been used to evaluate the effectiveness of information retrieval. Because there were many queries for which the average precision was zero in all methods, we discarded those queries. There were 686 remaining KQs and 273 remaining LQs.

The average number of relevant documents for each query was 2.1. Although this number is small, the number of queries is large. Therefore, our evaluation result can be stable, as in evaluations for question answering (Voorhees and Tice, 2000).

We can derive the following points from Table 2. First, to clarify the effectiveness of the lemmatization in information retrieval, we compare "no lemmatization" with the other methods. Any lemmatization method improved the MAP for both KQ and LQ. Thus, lemmatization was effective for information retrieval in Mongolian. Second, we compare the MAP of our method with those of Sanduijav et al. (2005) and Khaltar et al. (2006). Our method was more effective than the method of Sanduijav et al. (2005) for both KQ and LQ. However, the difference between Khaltar et al. (2006) and our method was small for KQ and our method

Title: **Альбумин үйлвэрлэх дэвшилтэт технологи**
Author's name: **Дорж Дандий**
Keywords: **цусны ийлдэс, эхэс …**
Abstract: **Судалгааны ажлын тайлан 5, 10% ийн…**
Result: **Альбумины уусмал үйлдвэрлэх, сонгон …**

Figure 8: Example of technical abstract.

Table 2: MAP of lemmatization methods.

	Keyword query	List query
No lemmatization	0.2312	0.2766
Sanduijav et al. (2005)	0.2882	0.2834
Khaltar et al. (2006)	0.3134	0.3127
Our method	0.3149	0.3114
Correct lemmatization	0.3268	0.3187

was less effective than Khaltar et al.(2006) for LQ. This is because although we enhanced the lemmatization for verbs, adjectives, numerals, and loanwords, the effects were overshadowed by a large number of queries comprising conventional Mongolian nouns. Finally, our method did not outperform the method using the correct lemmatization.

We used the paired t-test for statistical testing, which investigates whether the difference in performance is meaningful or simply because of chance (Keen, 1992). Table 3 shows the results, in which "<" and "<<" indicate that the difference of two results was significant at the 5% and 1% levels, respectively, and "—" indicates that the difference of two results was not significant.

Looking at Table 3, the differences between no lemmatization and any lemmatization method, such as Sanduijav et al. (2005), Khaltar et al. (2006), our method, and correct lemmatization, were statistically significant in MAP for KQ. However, because the MAP value of no lemmatization was improved for LQ, the differences between no lemmatization and the lemmatization methods were less significant than those for KQ. The difference between Sanduijav et al. (2005) and our method was statistically significant in MAP for both KQ and LQ. However, the difference between Khaltar et al. (2006) and our method was not significant in MAP for both KQ and LQ. Although, the difference between our method and correct lemmatization was statistically significant in MAP for KQ, the difference was not significant in MAP for LQ.

Table 3: t-test result of the differences between lemmatization methods.

	Keyword query	List query
No lemmatization vs. Correct lemmatization	<<	<
No lemmatization vs. Sanduijav et al. (2005)	<<	—
No lemmatization vs. Khaltar et al. (2006)	<<	<
No lemmatization vs. Our method	<<	<
Sanduijav et al. (2005) vs. Our method	<<	<
Khaltar et al. (2006) vs. Our method	—	—
Our method vs. Correct lemmatization	<	—

6 Conclusion

In Modern Mongolian, a content word can potentially be inflected when concatenated with suffixes. Identifying the original forms of content words is crucial for natural language processing and information retrieval.

In this paper, we proposed a lemmatization method for Modern Mongolian. We enhanced the lemmatization method proposed by Khaltar et al. (2006). We targeted nouns, verbs, adjectives, and numerals. We also improved the lemmatization for loanwords.

We evaluated our lemmatization method experimentally. The accuracy of our method was higher than those of existing methods. We also applied our lemmatization method to information retrieval and improved the retrieval accuracy.

Future work includes using a part of speech tagger because the part of speech information is effective for lemmatization.

References

Vinsensius Berlian Vega S N and Stéphane Bressan. 2001. Indexing the Indonesian Web: Language identification and miscellaneous issues. *Tenth International World Wide Web Conference, Hong Kong*.

Eric Brill. 1997. Natural Language Processing Using Very Large Corpora. Kluwer Academic Press.

Johan Carlberger, Hercules Dalianis, Martin Hassel, and Ola Knutsson. 2001. Improving Precision in Information Retrieval for Swedish using Stemming. *Proceedings of NODALIDA '01 - 13th Nordic Conference on Computational Linguistics.*

Terumasa Ehara, Suzushi Hayata, and Nobuyuki Kimura. 2004. Mongolian morphological analysis using ChaSen. *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing,* pp. 709-712. (In Japanese).

Çuna F. Ekmekçioglu, Michael F. Lynch, and Peter Willett. 1996. Stemming and n-gram matching for term conflation in Turkish texts. *Information Research News,* Vol. 7, No. 1, pp. 2-6.

David A. Hull. 1996. Stemming algorithms – a case study for detailed evaluation. *Journal of the American Society for Information Science and Technology*, Vol. 47, No. 1, pp. 70-84.

Lauri Karttunen and Kenneth R. Beesley. 2003. Finite State Morphology. *CSLI Publications*. Stanford.

Micheal E. Keen. 1992. Presenting results of experimental retrieval comparisons. *Information Processing and Management*, Vol. 28, No. 4, pp. 491-502.

Badam-Osor Khaltar, Atsushi Fujii, and Tetsuya Ishikawa. 2006. Extracting loanwords from Mongolian corpora and producing a Japanese-Mongolian bilingual dictionary. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 657-664.

Tuomo Korenius, Jorma Laurikkala, Kalervo Järvelin, and Martti Juhola. 2004. Stemming and Lemmatization in the Clustering of Finnish Text Documents. *Proceedings of the thirteenth Association for Computing Machinery international conference on Information and knowledge management.* pp. 625-633.

Richard J. Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, Vol. 33, No. 1, pp. 159-174.

Leah S. Larkey, Lisa Ballesteros, and Margaret E. Connel. 2002. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 275-282.

Massimo Melucci and Nicola Orio. 2003. A Novel Method for Stemmer Generation Based on Hidden Markov Models. *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 131-138.

Mirko Popovič and Peter Willett. 1992. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science and Technology*, Vol. 43, No. 5, pp. 384-390.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, Vol. 14, No. 3, pp. 130-137.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1995. Okapi at TREC-3. *Proceedings of the Third Text REtrieval Conference, NIST Special Publication* 500-226. pp. 109-126.

Enkhbayar Sanduijav, Takehito Utsuro, and Satoshi Sato. 2005. Mongolian phrase generation and morphological analysis based on phonological and morphological constraints. *Journal of Natural Language Processing,* Vol. 12, No. 5, pp. 185-205. (In Japanese) .

Sock Y. Tai, Cheng O. Ong, and Noor A. Abdullah. 2000. On designing an automated Malaysian stemmer for the Malay language. *Proceedings of the fifth international workshop on information retrieval with Asian languages, Hong Kong,* pp. 207-208.

Bayarmaa Ts. 2002. Mongolian grammar for grades I-IV. (In Mongolian).

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a Question Answering Test Collection. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 200-207.

Jinxi Xu and Bruce W. Croft. 1998. Corpus-based stemming using co-occurrence of word variants. *ACM Transactions on Information Systems,* Vol. 16, No. 1, pp. 61-81.

An Empirical Comparison of Goodness Measures for Unsupervised Chinese Word Segmentation with a Unified Framework

Hai Zhao*and Chunyu Kit
Department of Chinese, Translation and Linguistics,
City University of Hong Kong,
83 Tat Chee Avenue, Kowloon, Hong Kong, China
Email: haizhao@cityu.edu.hk, ctckit@cityu.edu.hk

Abstract

This paper reports our empirical evaluation and comparison of several popular goodness measures for unsupervised segmentation of Chinese texts using Bakeoff-3 data sets with a unified framework. Assuming no prior knowledge about Chinese, this framework relies on a goodness measure to identify word candidates from unlabeled texts and then applies a generalized decoding algorithm to find the optimal segmentation of a sentence into such candidates with the greatest sum of goodness scores. Experiments show that description length gain outperforms other measures because of its strength for identifying short words. Further performance improvement is also reported, achieved by proper candidate pruning and by assemble segmentation to integrate the strengths of individual measures.

1 Introduction

Unsupervised Chinese word segmentation was explored in a number of previous works for various purposes and by various methods (Ge et al., 1999; Fu and Wang, 1999; Peng and Schuurmans, 2001; SUN et al., 2004; Jin and Tanaka-Ishii, 2006). However, various heuristic rules are often involved in most existing works, and there has not been a comprehensive comparison of their performance in a unified way with available large-scale "gold standard" data sets, especially, multi-standard ones since Bakeoff-1 [1].

In this paper we will propose a unified framework for unsupervised segmentation of Chinese text. Four existing approaches to unsupervised segmentations or word extraction are considered as its special cases, each with its own goodness measurement to quantify word likelihood. The output by each approach will be evaluated using benchmark data sets of Bakeoff-3[2] (Levow, 2006). Note that unsupervised segmentation is different from, if not more complex than, word extraction, in that the former must carry out the segmentation task for a text, for which a segmentation (decoding) algorithm is indispensable, whereas the latter only acquires a word candidate list as output (Chang and Su, 1997; Zhang et al., 2000).

2 Generalized Framework

We propose a generalized framework to unify the existing methods for unsupervised segmentation, assuming the availability of a list of word candidates each associated with a goodness for how likely it is to be a true word. Let $W = \{\{w_i, g(w_i)\}_{i=1,...,n}\}$ be such a list, where w_i is a word candidate and $g(w_i)$

*The research described in this paper was supported by the Research Grants Council of Hong Kong S.A.R., China, through the CERG grant 9040861 (CityU 1318/03H) and by City University of Hong Kong through the Strategic Research Grant 7002037. Dr. Hai Zhao was supported by a postdoctoral Research Fellowship in the Department of Chinese, Translation and Linguistics, City University of Hong Kong. Thanks four anonymous reviewers for their insightful comments!

[1] First International Chinese Word Segmentation Bakeoff, at http://www.sighan.org/bakeoff2003

[2] The Third International Chinese Language Processing Bakeoff, at http://www.sighan.org/bakeoff2006.

its goodness function.

Two generalized decoding algorithms, (1) and (2), are formulated for optimal segmentation of a given plain text. The first one, decoding algorithm (1), is a Viterbi-style one to search for the best segmentation S^* for a text T, as follows,

$$S^* = \underset{w_1 \cdots w_i \cdots w_n = T}{\operatorname{argmax}} \sum_{i=1}^{n} g(w_i), \qquad (1)$$

with all $\{w_i, g(w_i)\} \in W$.

Another algorithm, decoding algorithm (2), is a maximal-matching one with respect to a goodness score. It works on T to output the best current word w^* repeatedly with T=t^* for the next round as follows,

$$\{w^*, t^*\} = \underset{wt = T}{\operatorname{argmax}}\ g(w) \qquad (2)$$

with each $\{w, g(w)\} \in W$. This algorithm will back off to forward maximal matching algorithm if the goodness function is set to word length. Thus the former may be regarded as a generalization of the latter. Symmetrically, it has an inverse version that works the other way around.

3 Goodness Measurement

An unsupervised segmentation strategy has to rest on some predefined criterion, e.g., mutual information (MI), in order to recognize a substring in the text as a word. Sproat and Shih (1990) is an early investigation in this direction. In this study, we examine four types of goodness measurement for a candidate substring[3]. In principle, the higher goodness score for a candidate, the more possible it is to be a true word.

Frequency of Substring with Reduction A linear algorithm was proposed in (Lü et al., 2004) to produce a list of such reduced substrings for a given corpus. The basic idea is that if two partially overlapped n-grams have the same frequency in the input corpus, then the shorter one is discarded as a redundant word candidate. We take the logarithm of FSR as the goodness for a word candidate, i.e.,

$$g_{FSR}(w) = \log(\hat{p}(w)) \qquad (3)$$

where $\hat{p}(w)$ is w's frequency in the corpus. This allows the arithmetic addition in (1). According to Zipf's Law (Zipf, 1949), it approximates the use of the rank of w as its goodness, which would give it some statistical significance. For the sake of efficiency, only those substrings that occur more than once are considered qualified word candidates.

Description Length Gain (DLG) The goodness measure is proposed in (Kit and Wilks, 1999) for compression-based unsupervised segmentation. The DLG from extracting all occurrences of $x_i x_{i+1} ... x_j$ (also denoted as $x_{i..j}$) from a corpus $X = x_1 x_2 ... x_n$ as a word is defined as

$$DLG(x_{i..j}) = L(X) - L(X[r \rightarrow x_{i..j}] \oplus x_{i..j}) \quad (4)$$

where $X[r \rightarrow x_{i..j}]$ represents the resultant corpus from replacing all instances of $x_{i..j}$ with a new symbol r throughout X and $\oplus$ denotes the concatenation of two substrings. $L(\cdot)$ is the empirical description length of a corpus in bits that can be estimated by the Shannon-Fano code or Huffman code as below, following classic information theory (Shannon, 1948).

$$L(X) \doteq -|X| \sum_{x \in V} \hat{p}(x) \log_2 \hat{p}(x) \qquad (5)$$

where $|\cdot|$ denotes string length, V is the character vocabulary of X and $\hat{p}(x)$ x's frequency in X. For a given word candidate w, we define $g_{DLG}(w) = DLG(w)$. In principle, a substring with a negative DLG do not bring any positive compression effect by itself. Thus only substrings with a positive DLG value are added into our word candidate list.

Accessor Variety (AV) Feng et al. (2004) propose AV as a statistical criterion to measure how likely a substring is a word. It is reported to handle low-frequent words particularly well. The AV of a substring $x_{i..j}$ is defined as

$$AV(x_{i..j}) = \min\{L_{av}(x_{i..j}), R_{av}(x_{i..j})\} \qquad (6)$$

where the left and right accessor variety $L_{av}(x_{i..j})$ and $R_{av}(x_{i..j})$ are, respectively, the number of distinct predecessor and successor characters. For a similar reason as to FSR, the logarithm of AV is used

[3] Although there have been many existing works in this direction (Lua and Gan, 1994; Chien, 1997; Sun et al., 1998; Zhang et al., 2000; SUN et al., 2004), we have to skip the details of comparing MI due to the length limitation of this paper. However, our experiments with MI provide no evidence against the conclusions in this paper.

as goodness measure, and only substrings with AV > 1 are considered word candidates. That is, we have $g_{AV}(w) = \log AV(w)$ for a word candidate w.

Boundary Entropy (Branching Entropy, BE) It is proposed as a criterion for unsupervised segmentation in some existing works (Tung and Lee, 1994; Chang and Su, 1997; Huang and Powers, 2003; Jin and Tanaka-Ishii, 2006). The local entropy for a given $x_{i..j}$, defined as

$$h(x_{i..j}) = -\sum_{x \in V} p(x|x_{i..j}) \log\, p(x|x_{i..j}), \qquad (7)$$

indicates the average uncertainty after (or before) $x_{i..j}$ in the text, where $p(x|x_{i..j})$ is the co-occurrence probability for x and $x_{i..j}$. Two types of $h(x_{i..j})$, namely $h_L(x_{i..j})$ and $h_R(x_{i..j})$, can be defined for the two directions to extend $x_{i..j}$ (Tung and Lee, 1994). Also, we can define $h_{min} = \min\{h_R, h_L\}$ in a similar way as in (6). In this study, only substrings with BE > 0 are considered word candidates. For a candidate w, we have $g_{BE}(w) = h_{min}(w)$[4].

4 Evaluation

The evaluation is conducted with all four corpora from Bakeoff-3 (Levow, 2006), as summarized in Table 1 with corpus size in number of characters. For unsupervised segmentation, the annotation in the training corpora is not used. Instead, they are used for our evaluation, for they are large and thus provide more reliable statistics than small ones. Segmentation performance is evaluated by word F-measure $F = 2RP/(R + P)$. The recall R and precision P are, respectively, the proportions of the correctly segmented words to all words in the gold-standard and a segmenter's output[5].

Note that a decoding algorithm always requires the goodness score of a single-character candidate for computation. There are two ways to get this score: (1) computed by the goodness measure, which is applicable only if the measure allows; (2) set to zero as default value, which is always applicable even to single-character candidates not in the word candidate list in use. For example, all single-character candidates given up by DLG because of their negative DLG scores will have a default value during decoding. We will use a '/d' to indicate experiments using such a default value.

Table 1: Bakeoff-3 Corpora

Corpus	AS	CityU	CTB	MSRA
Training(M)	8.42	2.71	0.83	2.17
Test(K)	146	364	256	173

Table 2: Performance with decoding algorithm (1)

M. L.[a]	Good-ness	Training corpus			
		AS	CityU	CTB	MSRA
2	FSR	.400	.454	.462	.432
	DLG/d	**.592**	**.610**	**.604**	**.603**
	AV	.568	.595	.596	.577
	BE	.559	.587	.592	.572
7	FSR	.193	.251	.268	.235
	DLG/d	.331	.397	.409	.379
	AV	**.399**	**.423**	**.430**	**.407**
	BE	.390	.419	.428	.403

[a]M.L.: Maximal length allowable for word candidates.

4.1 Comparison

We apply the decoding algorithm (1) to segment all Bakeoff-3 corpora with the above goodness measures. Both word candidates and goodness values are derived from the raw text of each training corpus. The performance of these measures is presented in Table 2. From the table we can see that DLG and FSR have the strongest and the weakest performance, respectively, whereas AV and BE are highly comparable to each other.

Decoding algorithm (2) runs the forward and backward segmentation with the respective AV and BE criteria, i.e., L_{AV}/h_L for backward and R_{AV}/h_R forward, and the output is the union of two segmentations [6]. A performance comparison of AV and BE with both algorithms (1) and (2) is presented in Table 3. We can see that the former has a rela-

[4]Both AV and BE share a similar idea from Harris (1970): If the uncertainty of successive token increases, then it is likely to be at a boundary. In this sense, one may consider them the discrete and continuous formulation of the same idea.

[5]All evaluations will be represented in terms of word F-measure if not otherwise specified. A standard scoring tool with this metric can be found in SIGHAN website, http://www.sighan.org/bakeoff2003/score. However, to compare with related work, we will also adopt boundary F-measure $F_b = 2R_bP_b/(R_b + P_b)$, where the boundary recall R_b and boundary precision P_b are, respectively, the proportions of the correctly recognized boundaries to all boundaries in the gold-standard and a segmenter's output (Ando and Lee, 2000).

[6]That is, all segmented points by either segmentation will be accounted into the final segmentation.

Table 3: Performance comparison: AV vs. BE

M. L.	Goodness	Training corpus AS	CityU	CTB	MSRA
2	$AV_{(1)}$	**.568**	**.595**	**.596**	**.577**
	$AV_{(2)}$/d	.485	.489	.508	.471
	$AV_{(2)}$	.445	.366	.367	.387
	$BE_{(1)}$	**.559**	**.587**	**.592**	**.572**
	$BE_{(2)}$/d	.485	.489	.508	.471
	$BE_{(2)}$	.504	.428	.446	.446
7	$AV_{(1)}$	.399	.423	.430	.407
	$AV_{(2)}$/d	**.570**	**.581**	**.588**	**.572**
	$AV_{(2)}$	.445	.366	.368	.387
	$BE_{(1)}$	.390	.419	.428	.403
	$BE_{(2)}$/d	**.597**	**.604**	**.605**	**.593**
	$BE_{(2)}$	.508	.431	.449	.446

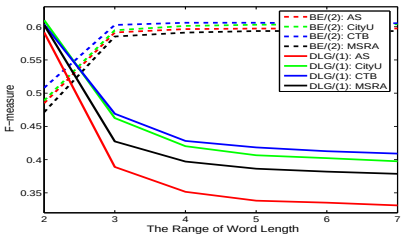


Figure 1: Performance vs. word length

tively better performance on shorter words and the latter outperforms on longer ones.

How segmentation performance varies along with word length is exemplified with DLG and BE as examples in Figure 1, with (1) and (2) indicating a respective decoding algorithm in use. It shows that DLG outperforms on two-character words and BE on longer ones.

4.2 Word Candidate Pruning

Up to now, word candidates are determined by the default goodness threshold 0. The number of them for each of the four goodness measures is presented in Table 4. We can see that FSR generates the largest set of word candidates and DLG the smallest. More interestingly or even surprising, AV and BE generate exactly the same candidate list for all corpora.

In addition to word length, another crucial factor to affect segmentation performance is the quality of the word candidates as a whole. Since each candidate is associated with a goodness score to indicate how good it is, a straightforward way to ensure, and further enhance, the overall quality of a candidate set is to prune off those with low goodness scores.

Table 4: Word candidate number by threshold 0

Goodness	Training Corpus AS	CityU	CTB	MSRA
FSR	2,009K	832K	294K	661K
DLG	543K	265K	96K	232K
AV	1,153K	443K	160K	337K
BE	1,153K	443K	160K	337K

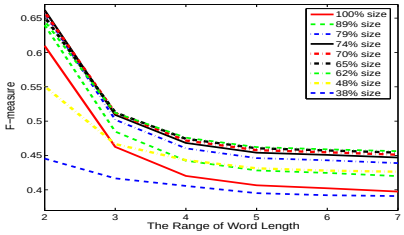


Figure 2: Performance by candidate pruning: DLG

To examine how segmentation performance changes along with word candidate pruning and decide the optimal pruning rate, we conduct a series of experiments with each goodness measurements. Figures 2 and 3 present, as an illustration, the outcomes of two series of our experiments with DLG by decoding algorithm (1) and BE by decoding algorithm (1) and (2) on CityU training corpus. We find that appropriate pruning does lead to significant performance improvement and that both DLG and BE keep their superior performance respectively on two-character words and others. We also observe that each goodness measure has a stable and similar performance in a range of pruning rates around the optimal one, e.g., 79-62% around 70% in Figure 2.

The optimal pruning rates found through our experiments for the four goodness measures are given in Table 5, and their correspondent segmentation performance in Table 6. These results show a remarkable performance improvement beyond the de-

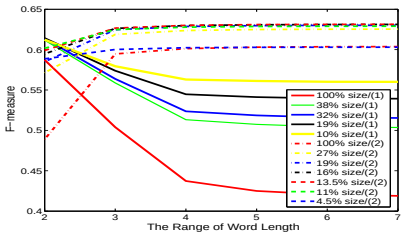


Figure 3: Performance by candidate pruning: BE

Table 5: Optimal rates for candidate pruning (%)

Decoding algorithm	Goodness measure			
	FSR	DLG	AV	BE
(1)	1.8	70	12.5	20
(2)	–	–	8	12.5

Table 6: Performance via optimal candidate pruning

M. L.	Good-ness	Training corpus			
		AS	CityU	CTB	MSRA
2	$FSR_{(1)}$	.501	.525	.513	.522
	$DLG_{(1)}$/d	**.710**	**.650**	**.664**	**.638**
	$AV_{(1)}$	.616	.625	.609	.618
	$BE_{(1)}$	.613	.614	.605	.611
	$AV_{(2)}$/d	.585	.602	.589	.599
	$BE_{(2)}$/d	.591	.599	.596	.593
7	$FSR_{(1)}$	.444	.491	.486	.486
	$DLG_{(1)}$/d	.420	.447	.460	.423
	$AV_{(1)}$	.517	.568	.549	.544
	$BE_{(1)}$	.501	.539	.510	.519
	$AV_{(2)}$/d	.623	.624	.604	.615
	$BE_{(2)}$/d	**.630**	**.631**	**.620**	**.622**

fault threshold setting. What remains unchanged is the advantage of DLG for two-character words and that of AV/BE for longer words. However, DLG achieves the best overall performance among the four, although it uses only single- and two-character word candidates. The overwhelming number of two-character words in Chinese allows it to triumph.

4.3 Ensemble Segmentation

Although proper pruning of word candidates brings amazing performance improvement, it is unlikely for one to determine an optimal pruning rate in practice for an unlabeled corpus. Here we put forth a parameter-free method to tackle this problem with the aids of all available goodness measures.

The first step of this method to do is to derive an optimal set of word candidates from the input. We have shown above that quality candidates play a critical role in achieving quality segmentation. Without any better goodness criterion available, the best we can opt for is the intersection of all word candidate lists generated by available goodness measures with the default threshold. A good reason for this is that the agreement of them can give a more reliable decision than any individual one of them. In fact, we only need DLG and AV/BE to get this intersection, because AV and BE give the same word candidates

Table 7: Performances of ensemble segmentation

M. L.	Good-ness	Training corpus			
		AS	CityU	CTB	MSRA
2	$FSR_{(1)}$	.629	.635	.624	.623
	$DLG_{(1)}$/d	**.664**	**.653**	**.643**	**.650**
	$AV_{(1)}$	.641	.644	.631	.634
	$BE_{(1)}$	.640	.643	.632	.634
7	$AV_{(2)}$/d	.595	.637	.624	.610
	$BE_{(2)}$/d	.593	.635	.620	.609
$DLG_{(1)}$/d+$AV_{(2)}$/d		**.672**	**.684**	**.663**	**.665**
$DLG_{(1)}$/d+$BE_{(2)}$/d		.660	.681	.656	.653

and DLG generates only a subset of what FSR does.

The next step is to use this intersection set of word candidates to perform optimal segmentation with each goodness measures, to see if any further improvement can be achieved. The best results are given in Table 7, showing that decoding algorithm (1) achieves marvelous improvement using short word candidates with all other goodness measures than DLG. Interestingly, DLG still remains at the top by performance despite of some slip-back.

To explore further improvement, we also try to combine the strengths of DLG and AV/BE respectively for recognizing two- and multi-character word. Our strategy to combine them together is to enforce the multi-character words in AV/BE segmentation upon the correspondent parts of DLG segmentation. This ensemble method gives a better overall performance than all others that we have tried so far, as presented at the bottom of Table 7.

4.4 Yet Another Decoding Algorithm

Jin and Tanaka-Ishii (2006) give an unsupervised segmentation criterion, henceforth referred to as decoding algorithm (3), to work with BE. It works as follows: if $g(x_{i..j+1}) > g(x_{i..j})$ for any two overlapped substrings $x_{i..j}$ and $x_{i..j+1}$, then a segmenting point should be located right after $x_{i..j+1}$. This algorithm has a forward and a backward version. The union of the segmentation outputs by both versions is taken as the final output of the algorithm, in exactly the same way as how decoding algorithm (2) works[7]. This algorithm is evaluated in (Jin and Tanaka-Ishii, 2006) using Peking University (PKU)

[7]Three segmentation criteria are given in (Jin and Tanaka-Ishii, 2006), among which the entropy increase criterion, namely, decoding algorithm (3), proves to be the best. Here we would like to thank JIN Zhihui and Prof. Kumiko Tanaka-Ishii for presenting the details of their algorithms.

Table 8: Performance comparison by word and boundary F-measure on PKU corpus (M. L. = 6)

	Good-ness	Decoding algorithm					
		(1)/d	(1)	(2)/d	(2)	(3)/d	(3)
F	AV	.313	.325	.588	.373	.376	.453
	AV*	.372	.372	**.663**	.663	.445	.445
	BE	.309	.319	.624	.501	.376	.624
	BE*	.370	.370	**.676**	.676	.447	.447
F_b	AV	.695	.700	.830	.762	.762	.728
	AV*	.728	.728	**.865**	.865	.783	.783
	BE	.696	.699	.849	.810	.762	.837[a]
	BE*	.728	.728	**.872**	.872	.784	.784

[a]With the same hyperparameters, (Jin and Tanaka-Ishii, 2006) report their best result of boundary precision 0.88 and boundary recall 0.79, equal to boundary F-measure 0.833.

Corpus of 1.1M words[8] as gold standard with a word candidate list extracted from the 200M Contemporary Chinese Corpus that mostly consists of several years of Peoples' Daily[9]. Here, we carry out evaluation with similar data: we extract word candidates from the unlabeled texts of People's Daily (1993 - 1997), of 213M and about 100M characters, in terms of the AV and BE criteria, yielding a list of 4.42 million candidates up to 6-character long[10] for each criterion. Then, the evaluation of the three decoding algorithms is performed on PKU corpus.

The evaluation results with both word and boundary F-measure are presented for the same segmentation outputs in Table 8, with "*" to indicate candidate pruning by DLG > 0 as reported before. Note that boundary F-measure gives much more higher score than word F-measure for the same segmentation output. However, in either of metric, we can find no evidence in favor of decoding algorithm (3). Undesirably, this algorithm does not guarantee a stable performance improvement with the BE measure through candidate pruning.

4.5 Comparison against Supervised Segmentation

Huang and Zhao (2007) provide empirical evidence to estimate the degree to which the four segmentation standards involved in the Bakeoff-3 differ from each other. As quoted in Table 9, a consistency rate beyond 84.8% is found among the four standards. If we do not over-expect unsupervised segmentation to achieve beyond what these standards agree with each other, it is reasonable to take this figure as the topline for evaluation. On the other hand, Zhao et al. (2006) show that the words of 1 to 2 characters long account for 95% of all words in Chinese texts, and single-character words alone for about 50%. Thus, we can take the result of the brute-force guess of every single character as a word as a baseline.

Table 9: Consistency rate among Bakeoff-3 segmentation standards (Huang and Zhao, 2007)

Test corpus	Training corpus			
	AS	CityU	CTB	MSRA
AS	1.000	0.926	0.959	0.858
CityU	0.932	1.000	0.935	0.849
CTB	0.942	0.910	1.000	0.877
MSRA	0.857	0.848	0.887	1.000

To compare to supervised segmentation, which usually involves training using an annotated training corpus and, then, evaluation using test corpus, we carry out unsupervised segmentation in a comparable manner. For each data track, we first extract word candidates from both the training and test corpora, all unannotated, and then evaluate the unsupervised segmentation with reference to the gold-standard segmentation of the test corpus. The results are presented in Table 10, together with best and worst official results of the Bakeoff closed test. This comparison shows that unsupervised segmentation cannot compete against supervised segmentation in terms of performance. However, the experiments generate positive results that the best combination of the four goodness measures can achieve an F-measure in the range of 0.65-0.7 on all test corpora in use without using any prior knowledge, but extracting word candidates from the unlabeled training and test corpora in terms of their goodness scores.

5 Discussion: How Things Happen

Note that DLG criterion is to perform segmentation with the intension to maximize the compression effect, which is a global effect through the text. Thus it works well incorporated with a probability maximization framework, where high frequent but independent substrings are effectively extracted and re-

[8]http://icl.pku.edu.cn/icl_groups/corpus/dwldform1.asp

[9]http://ccl.pku.edu.cn:8080/ccl_corpus/jsearch/index.jsp

[10]This is to keep consistence with (Jin and Tanaka-Ishii, 2006), where 6 is set as the maximum n-gram length.

Table 10: Comparison of performances against supervised segmentation

Type		Test corpus			
		AS	CityU	CTB	MSRA
Baseline		.389	.345	.337	.353
2	$DLG_{(1)}/d$	.597	.616	.601	.602
	$DLG^{*}_{(1)}/d$	.655	.659	.632	.655
	$AV_{(1)}$	.577	.603	.597	.583
	$AV^{*}_{(1)}$	.630	.650	.618	.638
	$BE_{(1)}$	.570	.598	.594	.580
	$BE^{*}_{(1)}$	.629	.649	.618	.638
7	$AV_{(2)}/d$	.512	.551	.543	.526
	$AV^{*}_{(2)}/d$	.591	.644	.618	.604
	$BE_{(2)}/d$	.518	.554	.546	.533
	$BE^{*}_{(2)}/d$	.587	.641	.614	.605
$DLG^{*}_{(1)}/d + AV^{*}_{(2)}/d$		**.663**	**.692**	**.658**	**.667**
$DLG^{*}_{(1)}/d + BE^{*}_{(2)}/d$		.650	.689	.650	.656
Worst closed		.710	.589	0.818	.819
Best closed		.958	.972	0.933	.963

combined. We know that most unsupervised segmentation criteria will bring up long word bias problem, so does DLG measure. This explains why it gives the worse results as long candidates are added.

As for AV and BE measures, both of them give the metric of the uncertainty before or after the current substring. This means that they are more concerned with local uncertainty information near the current substring, instead of global information among the whole text as DLG. Thus local greedy search in maximal matching style is more suitable for these two measures than Viterbi search.

Our empirical results about word candidate list with default threshold 0, where the same list is from AV and BE, give another proof that both AV and BE reflect the same uncertainty. The only difference is behind the fact that the former and the latter is in the discrete and continuous formulation, respectively.

6 Conclusion and Future Work

This paper reported our empirical comparison of a number of goodness measures for unsupervised segmentation of Chinese texts with the aid two generalized decoding algorithms. We learn no previous work by others for a similar attempt. The comparison is carried out with Bakeoff-3 data sets, showing that all goodness measures exhibit their strengths for recognizing words of different lengths and achieve a performance far beyond the baseline. Among them, DLG with decoding algorithm (1) can achieve the best segmentation performance for single- and two-character words identification and the best overall performance as well. Our experiments also show that the quality of word candidates plays a critical role in ensuring segmentation performance [11]. Proper pruning of candidates with low goodness scores to enhance this quality enhances the segmentation performance significantly. Also, the success of unsupervised segmentation depends strongly on an appropriate decoding algorithm. Generally, Viterbi-style decoding produces better results than best-first maximal-matching. But the latter is not shy from exhibiting its particular strength for identifying multi-character words.

Finally, the ensemble segmentation we put forth to combine the strengths of different goodness measures proves to be a remarkable success. It achieves an impressive performance improvement on top of individual goodness measures.

As for future work, it would be natural for researchers to enhance supervised learning for Chinese word segmentation with goodness measures introduced here. There does be two successful examples in our existing work (Zhao and Kit, 2007). This is still an ongoing work.

References

Rie Kubota Ando and Lillian Lee. 2000. Mostly-unsupervised statistical segmentation of Japanese: Applications to kanji. In *Proceedings of the first Conference on North American Chapter of the Association for Computational Linguistics and the 6th Conference on Applied Natural Language Processing*, pages 241–248, Seattle, Washington, April 30.

Jing-Shin Chang and Keh-Yih Su. 1997. An unsupervised iterative method for Chinese new lexicon extraction. *Computational Linguistics and Chinese Language Processing*, 2(2):97–148.

Lee-Feng Chien. 1997. PAT-tree-based keyword extraction for Chinese information retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–58, Philadelphia.

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, 30(1):75–93.

[11] This observation is shared by other researchers, e.g., (Peng et al., 2002).

Guo-Hong Fu and Xiao-Long Wang. 1999. Unsupervised Chinese word segmentation and unknown word identification. In *5th Natural Language Processing Pacific Rim Symposium 1999 (NLPRS'99), "Closing the Millennium"*, pages 32–37, Beijing, China, November 5-7.

Xianping Ge, Wanda Pratt, and Padhraic Smyth. 1999. Discovering Chinese words from unsegmented text. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–272, Berkeley, CA, USA, August 15-19. ACM.

Zellig Sabbetai Harris. 1970. Morpheme boundaries within words. In *Papers in Structural and Transformational Linguistics*, page 68 ‐ 77.

Jin Hu Huang and David Powers. 2003. Chinese word segmentation based on contextual entropy. In Dong Hong Ji and Kim-Ten Lua, editors, *Proceedings of the 17th Asian Pacific Conference on Language, Information and Computation*, pages 152–158, Sentosa, Singapore, October, 1-3. COLIPS Publication.

Chang-Ning Huang and Hai Zhao. 2007. Chinese word segmentation: A decade review. *Journal of Chinese Information Processing*, 21(3):8–20.

Zhihui Jin and Kumiko Tanaka-Ishii. 2006. Unsupervised segmentation of Chinese text by use of branching entropy. In *COLING/ACL 2006*, pages 428–435, Sidney, Australia.

Chunyu Kit and Yorick Wilks. 1999. Unsupervised learning of word boundary with description length gain. In M. Osborne and E. T. K. Sang, editors, *CoNLL-99*, pages 1–6, Bergen, Norway.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia, July.

Xueqiang Lü, Le Zhang, and Junfeng Hu. 2004. Statistical substring reduction in linear time. In Keh-Yih Su et al., editor, *Proceeding of the 1st International Joint Conference on Natural Language Processing (IJCNLP-2004)*, volume 3248 of *Lecture Notes in Computer Science*, pages 320–327, Sanya City, Hainan Island, China, March 22-24. Springer.

Kim-Teng Lua and Kok-Wee Gan. 1994. An application of information theory in Chinese word segmentation. *Computer Processing of Chinese and Oriental Languages*, 8(1):115–123.

Fuchun Peng and Dale Schuurmans. 2001. Self-supervised Chinese word segmentation. In *The Fourth International Symposium on Intelligent Data Analysis*, pages 238–247, Lisbon, Portugal, September, 13-15.

Fuchun Peng, Xiangji Huang, Dale Schuurmans, Nick Cercone, and Stephen Robertson. 2002. Using self-supervised word segmentation in Chinese information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 349–350, Tampere, Finland, August, 11-15.

Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October.

Richard Sproat and Chilin Shih. 1990. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4(4):336–351.

Maosong Sun, Dayang Shen, and Benjamin K. Tsou. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *COLING-ACL '98, 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume 2, pages 1265–1271, Montreal, Quebec, Canada.

Mao Song SUN, Ming XIAO, and Benjamin K. Tsou. 2004. Chinese word segmentation without using dictionary based on unsupervised learning strategy (in Chinese) (基于无指导学习策略的无词表条件下的汉语自动分词). *Chinese Journal of Computers*, 27(6):736–742.

Cheng-Huang Tung and His-Jian Lee. 1994. Identification of unknown words from corpus. *Computational Proceedings of Chinese and Oriental Languages*, 8:131–145.

Jian Zhang, Jianfeng Gao, and Ming Zhou. 2000. Extraction of Chinese compound words – an experimental study on a very large corpus. In *Proceedings of the Second Chinese Language Processing Workshop*, pages 132–139, Hong Kong, China.

Hai Zhao and Chunyu Kit. 2007. Incorporating global information into supervised learning for Chinese word segmentation. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 66–74, Melbourne, Australia, September 19-21.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. Effective tag set selection in Chinese word segmentation via conditional random field modeling. In *Proceedings of the 20th Asian Pacific Conference on Language, Information and Computation*, pages 87–94, Wuhan, China, November 1-3.

George Kingsley Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA.

A Hybrid Approach to the Induction of Underlying Morphology

Michael Tepper
Department of Linguistics
University of Washington
Seattle, WA 98195
mtepper@u.washington.edu

Fei Xia
Department of Linguistics
University of Washington
Seattle, WA 98195
fxia@u.washington.edu

Abstract

We present a technique for refining a baseline segmentation and generating a plausible underlying morpheme segmentation by integrating hand-written rewrite rules into an existing state-of-the-art unsupervised morphological induction procedure. Performance on measures which consider surface-boundary accuracy and underlying morpheme consistency indicates this technique leads to improvements over baseline segmentations for English and Turkish word lists.

1 Introduction

1.1 Unsupervised Morphological Induction

The primary goal of unsupervised morphological induction (UMI) is the simultaneous induction of a reasonable morphological lexicon as well as an optimal segmentation of a corpus of words, given that lexicon. The majority of existing approaches employ statistical modeling towards this goal, but differ with respect to how they learn or refine the morphological lexicon. While some approaches involve lexical priors, either internally motivated or motivated by the minimal description length (MDL) criterion, some utilize heuristics. Pure maximum likelihood (ML) approaches may refine the lexicon with heuristics in lieu of explicit priors (Creutz and Lagus, 2004), or not make categorical refinements at all concerning which morphs are included, only probabilistic refinements through a hierarchical EM procedure (Peng and Schuurmans, 2001). Approaches that optimize the lexicon with respect to priors come in several flavors. There are basic maximum a priori (MAP) approaches that try to maximize the probability of the lexicon against linguistically motivated priors (Deligne and Bimbot, 1997; Snover and Brent, 2001; Creutz and Lagus, 2005). An alternative to MAP, MDL approaches use their own set of priors motivated by complexity theory. These studies attempt to minimize lexicon complexity (bit-length in crude MDL) while simultaneously minimizing the complexity (by maximizing the probability) of the corpus given the lexicon (de Marcken, 1996; Goldsmith, 2001; Creutz and Lagus, 2002).

Many of the approaches mentioned above utilize a simplistic unigram model of morphology to produce the segmentation of the corpus given the lexicon. Substrings in the lexicon are proposed as morphs within a word based on frequency alone, independently of phrase-, word- and morph-surroundings (de Marcken, 1996; Peng and Schuurmans, 2001; Creutz and Lagus, 2002). There are many approaches, however, which further constrain the segmentation procedure. The work by Creutz and Lagus (2004; 2005; 2006) constrains segmentation by accounting for morphotactics, first assigning mophotactic categories (prefix, suffix, and stem) to baseline morphs, and then seeding and refining an HMM using those category assignments. Other more structured models include Goldsmith's (2001) work which, instead of inducing morphemes, induces morphological signatures like {*ø*, *s*, *ed*, *ing*} for English regular verbs. Some techniques constrain possible analyses by employing approximations for morphological meaning or usage to prevent false derivations (like *singed* = *sing* + *ed*). There is work by Schone and Jurafsky (2000; 2001) where *meaning* is proxied by word- and morph-context, condensed via LSA. Yarowsky and Wicentowski (2000) and Yarowsky et al. (2001) use expectations on relative frequency of aligned inflected-word, stem pairs, as well as POS context features, both of which approximate some sort of meaning.

1.2 Allomorphy in UMI

Allomorphy, or allomorphic variation, is the process by which a morpheme varies (orthographically or

phonologically) in particular contexts, as constrained by a grammar.[1] To our knowledge, there is only handful of work within UMI attempting to integrate allomorphy into morpheme discovery. A notable approach is the Wordframe model developed by Wicentowski (2002), which performs weighted edits on root-forms, given context, as part of a larger similarity alignment model for discovering <inflected-form, root-form> pairs.

Morphological complexity is fixed by a template; the original was designed for inflectional morphologies and thus constrained to finding an optional affix on either side of a stem. Such a template would be difficult to design for agglutinative morphologies like Turkish or Finnish, where stems are regularly inflected by chains of affixes. Still, it can be extended. A notable recent extension accounts for phenomena like infixation and reduplication in Filipino (Cheng and See, 2006).

In terms of allomorphy, the approach succeeds at generalizing allomorphic patterns, both stem-internally and at points of affixation. A major drawback is that, so far, it does not account for affix allomorphy involving character replacement–that is, beyond point-of-affixation epentheses or deletions.

1.3 Our Approach

Our approach aims to integrate a rule-based component consisting of hand-written rewrite rules into an otherwise unsupervised morphological induction procedure in order to refine the segmentations it produces.

1.3.1 Context-Sensitive Rewrite Rules

The major contribution of this work is a rule-based component which enables simple encoding of context-sensitive rewrite rules for the analysis of induced morphs into plausible underlying morphemes.[2] A rule has the form general form:

$$\underset{\text{underlying}}{\alpha} \rightarrow \underset{\text{surface}}{\beta} \; / \; \underset{\text{l. context}}{\gamma} \_ \underset{\text{r. context}}{\delta} \qquad (1)$$

It is also known as a SPE-style rewrite rule, part of the formal apparatus to introduced by Chomsky and Halle (1968) to account for regularities in phonology. Here we use it to describe orthographic patterns. Mapping morphemes to underlying forms with context-sensitive rewrite rules allows us to peer through the fragmentation created by allomorphic variation. Our experiments will show that this has the effect of allowing for more unified, consistent morphemes while simultaneously making surface boundaries more transparent.

For example, take the English multipurpose inflectional suffix $\cdot s$, normally written as $\cdot s$, but as $\cdot es$ after sibilants (*s*,*sh*, *ch*, ...). We can write the following SPE-style rule to account for its variation.

$$\underset{\text{underlying}}{\emptyset} \rightarrow \underset{\text{surface}}{\text{e}} \; / \; [+\text{SIB}] + \_ s \qquad (2)$$

This rule says, "Insert an *e* (map *nothing* to *e*) following a character marked as a sibilant (+SIB) and a morphological boundary (+), at the focus position (_), immediately preceding an *s*." In short, it enables the mapping of the underlying form $\cdot s$ to $\cdot es$ by inserting an *e* before *s* where appropriate. When this rule is reversed to produce underlying analyses, the $\cdot es$ variant in such words as glasses, matches, swishes, and buzzes can be identified with the $\cdot s$ variant in words like plots, sits, quakes, and nips.

1.3.2 Overview of Procedure

Before the start of the procedure, there is a preprocessing step to derive an initial segmentation.

This segmentation is fed to the EM Stage, the goal of which is to find the maximum probability segmentation of a wordlist into *underlying* morphemes. First, analyses of initial segments are produced by rule. Then, their frequency is used to determine their likelihood as underlying morphemes. Finally, probability of a segmentation into underlying morphemes is maximized.

The output segmentation feeds into the Split Stage, where heuristics are used to split large, high-frequency segments that fail to break into smaller underlying morphemes during the EM algorithm.

2 Procedure

A flowchart of the procedure is given in Figure 1.

Preprocessing We use the Categories-MAP algorithm developed by Creutz and Lagus (2005; 2006) to produce an initial morphological segmentation. Here, a segmentation is optimized by maximum a posteriori estimate given priors on length, frequency, and usage of morphs stored in the model. Their procedure begins with morphological tags indicating basic morphotactics (prefix, stem, suffix, noise) being assigned heuristically to a baseline segmentation. That tag assignment is then used to seed an HMM.

[1] In this work we focus on orthographic allomorphy.

[2] Ordered rewrite rules, when restricted from applying to their own output, have similar expressive capabilities to Koskenniemi's two-level constraints. Both define regular relations on strings, both can be compiled into lexical transducers, and both have been used in finite-state analyzers (Karttunen and Beesley, 2001). We choose ordered rules because they are easier to write given our task and resources.

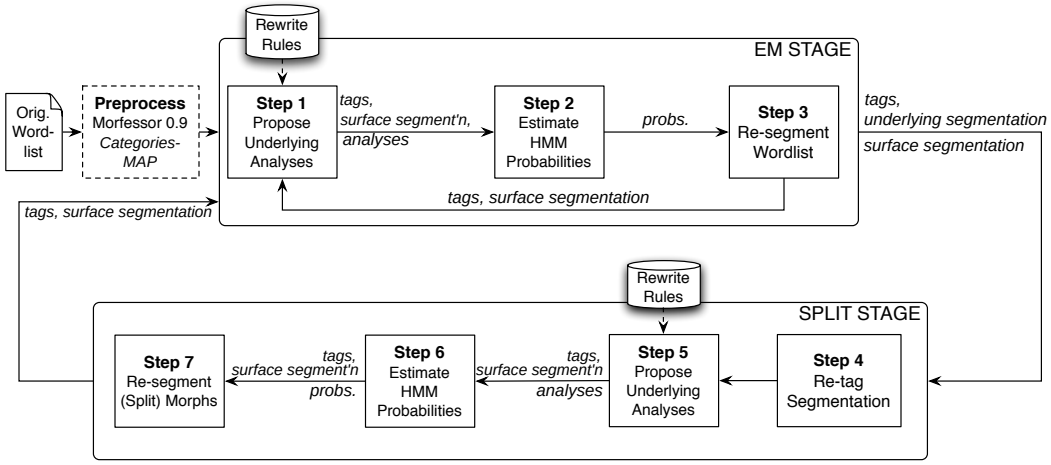


Figure 1: Flowchart showing the entire procedure.

Optimal segmentation of a word is simultaneously the best tag and morph[3] sequence given that word. The contents of the model are optimized with respect to length, frequency, and usage priors during splitting and joining phases. The final output is a tagged segmentation of the input word-list.

2.1 EM Stage

The model we train is a modified version of the morphological HMM from the work of Creutz and Lagus (2004-2006), where a word w consists of a sequence of morphs generated by a morphological-category tag sequence. The difference between their HMM and ours is that theirs emits surface morphs, while ours emits *underlying* morphemes. Morphemes may either be analyses proposed by rule or surface morphs acting as morphemes. We do not modify the tags Creutz and Lagus use (prefix, stem, suffix, and noise).

We proceed by EM, initialized by the preprocessed segmentation. Rule-generated underlying analyses are produced (Step 1), and used to estimate the emission probability $P(u_i|t_i)$ and transition probability $P(t_i|t_{i-1})$ (Step 2). In successive E-steps, Steps 1 and 2 are repeated. The M-step (Step 3) involves finding the maximum probability decoding of each word according to Eq (6), i.e. maximum probability tag and morpheme sequence.

Step 1 - Derive Underlying Analyses In this step, handwritten context-sensitive rewrite rules derive context-relevant analyses for morphs in the preprocessed segmentation. These analyses are produced by a set of ordered rules that propose deletions, insertions, or substitutions when triggered by the proper characters around a segmentation boundary.[4] A rule applies wherever contextually triggered, from left to right, and may apply more than once to the same word. To prevent the runaway application of certain rules, a rule may not apply to its own output. The result of applying a rule is a (possibly spelling-changed) segmented word, which is fed to the next rule. This enables multi-step analyses by using rules designed specifically to apply to the outputs of other rules. See Figure 2 for a small example.

Step 2 - Estimate HMM Probabilities Transition probabilities $P(t_i|t_{i-1})$ are estimated by maximum likelihood, given a tagged input segmentation.

Emission probabilities $P(u_i|t_i)$ are also estimated by maximum likelihood, but the situation is slightly more complex; the probability of morphemes u_i are estimated according to frequencies of association (coindexation) with surface morphs s_i and tags t_i.

Furthermore an underlying morpheme u_i can either be identical to its associated surface morph s_i when no rules apply, or be a rule-generated analysis. For the sake of clarity, we call the former u_i' and the latter u_i'', as defined below:

$$u_i = \begin{cases} u_i' & \text{if } u_i = s_i \\ u_i'' & \text{otherwise} \end{cases}$$

When an underlying morpheme u_i is associated to a surface morph s, we refer to s as an *allomorph* of

[3] A *morph* is a linguistic morpheme as it occurs in production, i.e. as it occurs in a *surface* word.

[4] Some special substitution rules, like vowel harmony in Turkish and Finnish, have a spreading effect, moving from syllable to syllable within and beyond morph-boundaries. In our formulation, these rules differ from other rules by not being conditioned on a morph-boundary.

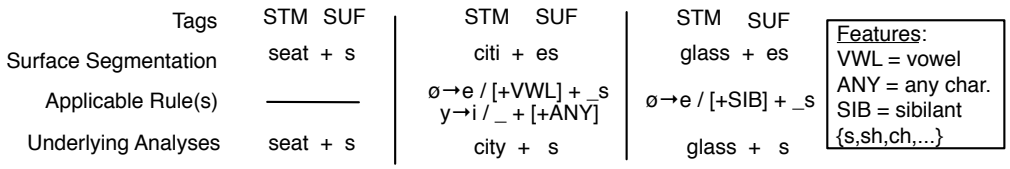


Figure 2: Underlying analyses for a segmentation are generated by passing it through context-sensitive rewrite rules. Rules apply to some morphs (e.g., *citi* → *city*) but not to others (e.g., *glass* → *glass*).

u_i. The probability of u_i given tag t_i is calculated by summing over all allomorphs s of u_i the probability that u_i realizes s in the context of tag t_i:

$$P(u_i|t_i) = \sum_{s \in \text{allom.-of}(u_i)} P(u_i, s|t_i) \quad (3)$$

$$= \sum_{s \in \text{allom.-of}(u_i)} P(u_i|s, t_i)P(s|t_i) \quad (4)$$

Both Eq (3) and Eq (4) are trivial to estimate with counting on our input from Step 1 (see Figure 2). We show (4) because it has the term $P(u_i|s, t_i)$, which may be used for thresholding and discounting terms of the sum where u_i is rarely associated with a particular allomorph and tag. In the future, such discounting may be useful to filter out noise generated by noisy or permissive rules. So far, this type of discounting has not improved results.

Step 3 - Resegment Word List Next we resegment the word list into underlying morphemes.

Searching for the best breakdown of a word w into morpheme sequence $\mathbf{u}$ and tag sequence $\mathbf{t}$, we maximize the probability of the following formula:

$$P(w, \mathbf{u}, \mathbf{t}) = P(w|\mathbf{u}, \mathbf{t})P(\mathbf{u}, \mathbf{t})$$

$$= P(w|\mathbf{u}, \mathbf{t})P(\mathbf{u}|\mathbf{t})P(\mathbf{t}) \quad (5)$$

To simplify, we assume that $P(w|\mathbf{u}, \mathbf{t})$ is equal to one.[5] With this assumption in mind, Eq (5) reduces to $P(\mathbf{u}|\mathbf{t})P(\mathbf{t})$. With independence assumptions and a local time horizon, we estimate:

$$\operatorname*{argmax}_{\mathbf{u},\mathbf{t}} P(\mathbf{u}|\mathbf{t})P(\mathbf{t})$$

$$\approx \operatorname*{argmax}_{\mathbf{u},\mathbf{t}} \Big[\prod_{i=1}^{n} P(u_i|t_i)P(t_i|t_{i-1})\Big] \quad (6)$$

[5]In other words, we make the assumption that a sequence of underlying morphemes and tags corresponds to just one word. This assumption may need revision in cases where morphemes can optionally undergo the types of spelling changes we are trying to encode; this has not been the case for the languages under investigation.

The search for the maximum probability tag and morph sequence in Eq (6) is carried out by a modified version of the Viterbi algorithm. The maximum probability segmentation for a given word may be a mixture of both types of underlying morpheme, u'_i and u''_i. Also, wherever we have a choice between emitting u'_i, identical to the surface form, or u''_i, an analysis with rule-proposed changes, the highest probability of the two is always selected.

2.2 Split Stage

Many times, large morphs have substructure and yet are too frequent to be split when segmented by the HMM in the EM Stage. To overcome this, we approximately follow the heuristic procedure[6] laid out by Creutz and Lagus (2004), encouraging splitting of larger morphs into smaller underlying morphemes. This process has the danger of introducing many false analyses, so first the segmentation must be re-tagged (Step 4) to identify which morphemes are noise and should not be used. Once we re-tag, we re-analyze morphs in the surface segmentation (Step 5) and re-estimate HMM probabilities (Step 6). (for Steps 5 and 6, refer to Steps 1 and 2). Finally, we use these HMM probabilities to split morphs (Step 7).

Step 4 - Re-tag the Segmentation To identify noise morphemes, we estimate a distribution $P(CAT|u_i)$ for three true categories CAT (prefix, stem, or suffix) and one noise category; we then assign categories randomly according to this distribution. Stem probabilities are proportional to stem-length, while affix probabilities are proportional to left- or right- perplexity. The probability of true categories are also tied to the value of sigmoid-cutoff parameters, the most important of which is b, which thresholds the probability of both types of affix (prefix and suffix).

The probability of the noise category is conversely related to the product of true category probabilities;

[6]The main difference between our procedure and Creutz and Lagus (2004) is that we allow splitting into two or more morphemes (see Step 7) while they allow binary splits only.

when true categories are less probable, noise becomes more probable. Thus, adjusting parameters like b can increase or decrease the probability of noise.

Step 7 - Split Morphs In this step, we examine <morph, tag> pairs in the segmentation to see if a split into sub-morphemes is warranted. We constrain this process by restricting splitting to stems (with the option to split affixes), and by splitting into restricted sequences of tags, particularly avoiding noise. We also use parameter b in Step 4 as a way to discourage excessive splitting by tagging more morphemes as noise. Stems are split into the sequence: (PRE* STM SUF*). Affixes (prefixes and suffixes) are split into other affixes of the same category. Whether to split affixes depends on typological properties of the language. If a language has agglutinative suffixation, for example, we hand-set a parameter to allow suffix-splitting.

When examining a morph for splitting, we search over all segmentations with at least one split, and choose the one that is both optimal according to Eq (6) and does not violate our constraints on what category sequences are allowed for *its* category. We end this step by returning to the EM Stage, where another cycle of EM is performed.

3 Experiments and Results

In this section we report and discuss development results for English and Turkish. We also report final-test results for both languages. Results for the pre-processed segmentation are consistently used as a baseline. In order to isolate the effect of the rewrite rules, we also compare against results taken on a parallel set of experiments, run with all the same parameters but without rule-generated underlying morphemes, i.e. without morphemes of type u_i''. But before we get to these results, we will describe the conditions of our experiments. First we introduce the evaluation metrics and data used, and then detail any parameters set during development.

3.1 Evaluation Metrics

We use *two* procedures for evaluation, described in the Morpho Challenge '05 and '07 Competition Reports (Kurimo et al., 2006; Kurimo et al., 2007). Both procedures use gold-standards created with commercially available morphological analyzers for each language. Each procedure is associated with its own F-score-based measure.

The first was used in Morpho Challenge '05, and measures the extent to which *boundaries* match between the surface-layer of our segmentations and gold-standard surface segmentations.

The second was used in Morpho Challenge '07 and measures the extent to which *morphemes* match between the underlying-layer of our segmentations and gold-standard underlying analyses. The F-score here is not actually on matched morphemes, but instead on matched morpheme-sharing word-pairs. A point is given whenever a morpheme-sharing word-pair in the gold-standard segmentation also shares morphemes in the test segmentation (for recall), and vice-versa for precision.

3.2 Data

Training Data The data-sets used for training were provided by the Helsinki University of Technology in advance of the Morpho Challenge '07 and were downloaded by the authors from the contest website[7]. According to the website, they were compiled from the University of Leipzig Wortschatz Corpora.

	Sentences	**Tokens**	**Types**
English	3×10^6	6.22×10^7	3.85×10^5
Turkish	1×10^6	1.29×10^7	6.17×10^5

Table 1: Training corpus sizes vary slightly, with 3 million English sentences and 1 million Turkish sentences.

Development Data The development gold-standard for the surface metric was provided in advance of Morpho Challenge '05 and consists of surface segmentations for 532 English and 774 Turkish words.

The development gold-standard for the underlying metric was provided in advance of Morpho Challenge '07 and consists of morphological analyses for 410 English and 593 Turkish words.

Test Data For final testing, we use the gold-standard data reserved for final evaluation in the Morpho Challenge '07 contest. The gold-standard consists of approximately 1.17×10^5 English and 3.87×10^5 Turkish analyzed words, roughly a tenth the size of training word-lists. Word pairs that exist in both the training and gold standard are used for evaluation.

3.3 Parameters

There are two sets of parameters used in this experiment. First, there are parameters used to produce the initial segmentation. They were set as suggested in Cruetz and Lagus (2005), with parameter b tuned on development data.

[7] http://www.cis.hut.fi/morphochallenge2007/datasets.shtml

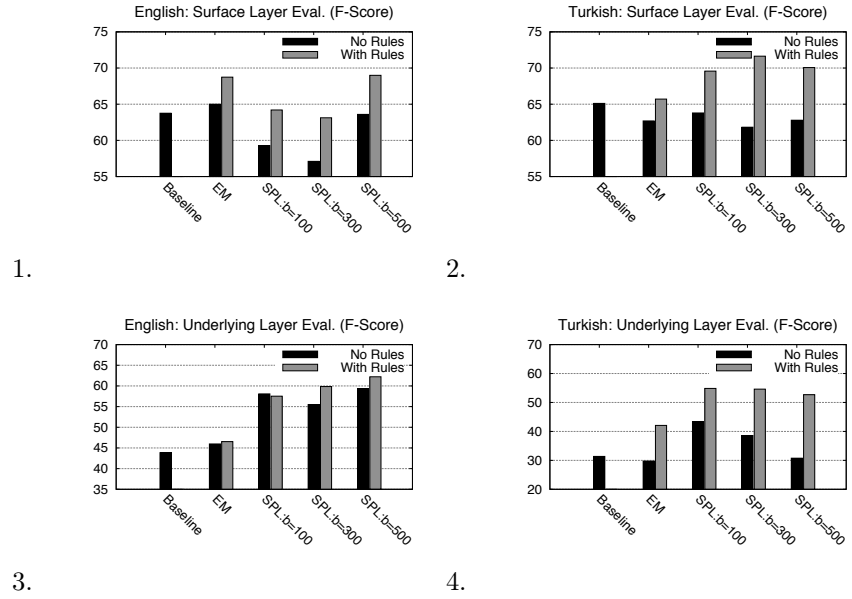


Figure 3: Development results for the preprocessed initial segmentation (Baseline), and segmentations produced by our approach, first after the EM Stage (EM) and again after the Split Stage (SPL) with different values of parameter b. Rules that generate underlying analyses have either been included (With Rules), or left out (No Rules).

Then there are parameters used for the main procedure. Here we have rewrite rules, numerical parameters, and one typology parameter. Rewrite rules and any orthographic features they use were culled from linguistic literature. We currently have 6 rules for English and 10 for Turkish; See Appendix A.1 for the full set of English rules used. Numerical parameters were set as suggested in Cruetz and Lagus (2004), and following their lead we tuned b on development data; we show development results for the following values: $b = 100$, 300, and 500 (see Figure 3). Finally, as introduced in Section 2.2, we have a hand-set typology parameter that allows us to split prefixes or suffixes if the language has an agglutinative morphology. Since Turkish has agglutinative suffixation, we set this parameter to split suffixes for Turkish.

3.4 Development Results

Development results were obtained by evaluating English and Turkish segmentations at several stages, and with several values of parameter b as shown in Figure 3.

Overall, our development results were very positive. For the surface-level evaluation, the largest F-score improvement was observed for English (Figure 3, Chart 1), 63.75% to 68.99%, a relative F-score gain of 8.2% over the baseline segmentation. The Turkish result also improves to a similar degree, but it is only achieved after the model as been refined by splitting. For English we observe the improvement earlier, after the EM Stage. For the underlying-level evaluation, the largest F-score improvement was observed for Turkish (Chart 4), 31.37% to 54.86%, a relative F-score gain of over 74%.

In most experiments with rules to generate underlying analyses (*With Rules*), the successive applications of EM and splitting result in improved results. Without rule-generated forms (*No Rules*) the results tend be negative compared to the baseline (see Figure 3, Chart 2), or mixed (Charts 1 and 4). When we look at recall and precision numbers directly, we observe that even without rules, the algorithm produces large recall boosts (especially after splitting). However, these boosts are accompanied by precision losses, which result in unchanged or lower F-scores.

The exception is the underlying-level evaluation of English segmentations (Figure 3, Chart 3). Here we observe a near-parity of F-score gains for segmentations produced with and without underlying morphemes derived by rule. One explanation is that the English initial segmentation is conservative and that coverage gains are the main reason for improved English scores. Creutz and Lagus (2005) note that the Morfessor EM approach often has better coverage than the MAP approach we use to produce the

				Hybrid:After Split	
	MC Morf.	**MC Top**	**Baseline**	**No Rules**	**With Rules**
English	47.17	**60.81**	47.04	57.35	59.78
Turkish	37.10	29.23	32.76	31.10	**54.54**

Table 2: Final test F-scores on the underlying morpheme measure used in Morpho Challenge '07. MC Morf. is Morfessor MAP, which was used as a reference method in the contest. MC Top is the top contestant. For our hybrid approach, we show the F-score obtained with and without using rewrite rules. The splitting parameter b was set to the best performing value seen in development evaluations (Tr. $b = 100$, En. $b = 500$).

initial segmentation. Also, in English, allomorphy is not as extensive as in Turkish (see Chart 4) where precision losses are greater without rules, i.e. when not representing allomorphs by the same morpheme.

3.5 Final Test Results

Final test results, given in Table 2, are mixed. For English, though we improve on our baseline and on Morfessor MAP trained by Creutz and Lagus, we are beaten by the top unsupervised Morpho Challenge contestant, entered by Delphine Bernhard (2007). Bernhard's approach was purely unsupervised and did not explicitly account for allomorphic phenomena. There are several possible reasons why we were not the top performer here. Our splitting constraint for stems, which allows them to split into stems and chains of affixes, is suited for agglutinative morphologies. It does not seem particularly well suited to English morphology. Our rewrite-rules might also be improved. Finally, there may be other, more pressing barriers (besides allomorphy) to improving morpheme induction in English, like ambiguity between homographic morphemes.

For Turkish, the story is very different. We observe our baseline segmentation going from 32.76% F-score to 54.54% when re-segmented using rules, a relative improvement of over 66%. Compared with the top unsupervised approach, Creutz and Lagus's Morfessor MAP, our F-score improvement is over 48%. The distance between our hybrid approach and unsupervised approaches emphasizes the problem allomorphy can be for a language like Turkish. Turkish inflectional suffixes, for instance, regularly undergo multiple spelling-rules and can have 10 or more variant forms. Knowing that these variants are all one morpheme makes a difference.

4 Conclusion

In this work we showed that we can use a small amount of knowledge in the form of context-sensitive rewrite rules to improve unsupervised segmentations for Turkish and English. This improvement can be quite large. On the morpheme-consistency measure used in the last Morpho Challenge, we observed an improvement of the Turkish segmentation of over 66% against the baseline, and 48% against the top-of-the-line unsupervised approach.

Work in progress includes error analysis of the results to more closely examine the contribution of each rule, as well as developing rule sets for additional languages. This will help highlight various aspects of the most beneficial rules.

There has been recent work on discovering allomorphic phenomena automatically (Dasgupta and Ng, 2007; Demberg, 2007). It is hoped that our work can inform these approaches, if only by showing what variation is possible, and what is relevant to particular languages. For example, variation in inflectional suffixes, driven by vowel harmony and other phenomena, should be captured for a language like Turkish.

Future work involves attempting to learn broad-coverage underlying morphology without the hand-coded element of the current work. This might involve employing aspects of the most beneficial rules as variable features in rule-templates. It is hoped that we can start to derive underlying morphemes through processes (rules, constraints, etc) suggested by these templates, and possibly learn instantiations of templates from seed corpora.

A Appendix

A.1 Rules Used For English

e epenthesis before s suffix
ø →e / ..[+V] + _s
ø→e / ..[+SIB] + _s
long e deletion
e →ø / ..[+V][+C]_ + [+V]
change y to i before suffix
y →i / ..[+C] +? _ + [+ANY]
consonant gemination
ø →α[+STOP] / ..α[+STOP]_ + [+V]
ø →α[+STOP] / ..α[+STOP]_ + [+GLI]

Table 3: English Rules

A.2 Example Segmentations

Base	EM	SPL:b=300	SPL:b=500
happen s	happen s	happ e n s	happen s
happier	happier	happi er	happi er
happiest	happiest	happ i est	happiest
happily	happily	happi ly	happi ly
happiness	happiness	happi ness	happiness

Table 4: Surface segmentations after preprocessing (Base), EM Stage (EM), and Split Stage (SPL)

References

Delphine Bernhard. 2007. Simple morpheme labeling in unsupervised morpheme analysis. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary.

Charibeth K. Cheng and Solomon L. See. 2006. The revised wordframe model for the filipino language. *Journal of Research in Science, Computing and Engineering*.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proc. Workshop on Morphological and Phonological Learning of ACL'02*, pages 21–30, Philadelphia. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2004. Induction of a simple morphology for highly inflecting languages. In *Proc. 7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 43–51, Barcelona.

Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proc. International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pages 106–113, Espoo, Finland.

Mathias Creutz and Krista Lagus. 2006. Morfessor in the morpho challenge. In *Proc. PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, Venice, Italy.

Sajib Dasgupta and Vincent Ng. 2007. High performance, language-independent morphological segmentation. In *Proc. NAACL'07*.

Carl G. de Marcken. 1996. *Unsupervised Language Acquisition*. Ph.D. thesis, Massachussetts Institute of Technology, Boston.

Sabine Deligne and Frédéric Bimbot. 1997. Inference of variable-length linguistic and acoustic units by multigrams. *Speech Communication*, 23:223–241.

Vera Demberg. 2007. A language-independent unsupervised model for morphological segmentation. In *Proc. ACL'07*.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27.2:153–198.

Lauri Karttunen and Kenneth R. Beesley. 2001. A short history of two-level morphology. In *Proc. ESSLLI 2001*.

Mikko Kurimo, Mathias Creutz, Matti Varjokallio, Ebru Arisoy, and Murat Saraçlar. 2006. Unsupervised segmentation of words into morphemes – Morpho Challenge 2005, an introduction and evaluation report. In *Proc. PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, Venice, Italy.

Mikko Kurimo, Mathias Creutz, and Matti Varjokallio. 2007. Unsupervised morpheme analysis evaluation by a comparison to a linguistic gold standard – Morpho Challenge 2007. In *Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary.

Fuchun Peng and Dale Schuurmans. 2001. A hierarchical em approach to word segmentation. In *Proc. 4th Intl. Conference on Intel. Data Analysis (IDA)*, pages 238–247.

Patrick Schone and Daniel Jurafsky. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Proc. CoNLL'00 and LLL'00*, pages 67–72, Lisbon.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Proc. NAACL'01*, Pittsburgh.

Matthew G. Snover and Michael R. Brent. 2001. A bayesian model for morpheme and paradigm identification. In *Proc. ACL'01*, pages 482–490, Toulouse, France.

Richard Wicentowski. 2002. *Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework*. Ph.D. thesis, Johns Hopkins University, Baltimore, Maryland.

David Yarowsky and Richard Wicentowski. 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proc. ACL'00*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection accross aligned corpora. In *Proc. HLT'01*, volume HLT 01, pages 161–168, San Diego.

Context-Sensitive Convolution Tree Kernel for Pronoun Resolution

ZHOU GuoDong KONG Fang ZHU Qiaoming
JiangSu Provincial Key Lab for Computer Information Processing Technology
School of Computer Science and Technology
Soochow Univ. Suzhou, China 215006
Email: {gdzhou, kongfang, qmzhu}@suda.edu.cn

Abstract

This paper proposes a context-sensitive convolution tree kernel for pronoun resolution. It resolves two critical problems in previous researches in two ways. First, given a parse tree and a pair of an anaphor and an antecedent candidate, it implements a dynamic-expansion scheme to automatically determine a proper tree span for pronoun resolution by taking predicate- and antecedent competitor-related information into consideration. Second, it applies a context-sensitive convolution tree kernel, which enumerates both context-free and context-sensitive sub-trees by considering their ancestor node paths as their contexts. Evaluation on the ACE 2003 corpus shows that our dynamic-expansion tree span scheme can well cover necessary structured information in the parse tree for pronoun resolution and the context-sensitive tree kernel much outperforms previous tree kernels.

1 Introduction

It is well known that syntactic structured information plays a critical role in many critical NLP applications, such as parsing, semantic role labeling, semantic relation extraction and co-reference resolution. However, it is still an open question on what kinds of syntactic structured information are effective and how to well incorporate such structured information in these applications.

Much research work has been done in this direction. Prior researches apply feature-based methods to select and define a set of flat features, which can be mined from the parse trees, to represent particular structured information in the parse tree, such as the grammatical role (e.g. subject or object), according to the particular application. Indeed, such feature-based methods have been widely applied in parsing (Collins 1999; Charniak 2001), semantic role labeling (Pradhan et al 2005), semantic relation extraction (Zhou et al 2005) and co-reference resolution (Lapin and Leass 1994; Aone and Bennett 1995; Mitkov 1998; Yang et al 2004; Luo and Zitouni 2005; Bergsma and Lin 2006). The major problem with feature-based methods on exploring structured information is that they may fail to well capture complex structured information, which is critical for further performance improvement.

The current trend is to explore kernel-based methods (Haussler, 1999) which can implicitly explore features in a high dimensional space by employing a kernel to calculate the similarity between two objects directly. In particular, the kernel-based methods could be very effective at reducing the burden of feature engineering for structured objects in NLP, e.g. the parse tree structure in coreference resolution. During recent years, various tree kernels, such as the convolution tree kernel (Collins and Duffy 2001), the shallow parse tree kernel (Zelenko et al 2003) and the dependency tree kernel (Culota and Sorensen 2004), have been proposed in the literature. Among previous tree kernels, the convolution tree kernel represents the state-of-the-art and have been successfully applied by Collins and Duffy (2002) on parsing, Moschitti (2004) on semantic role labeling, Zhang et al (2006) on semantic relation extraction and Yang et al (2006) on pronoun resolution.

However, there exist two problems in Collins and Duffy's kernel. The first is that the sub-trees enumerated in the tree kernel are context-free. That is, each sub-tree enumerated in the tree kernel does not consider the context information outside the sub-tree. The second is how to decide a proper tree span in the tree kernel computation according to the particular application. To resolve above two problems, this paper proposes a new tree span scheme and applies a new tree kernel and to better capture syntactic structured information in pronoun

resolution, whose task is to find the corresponding antecedent for a given pronominal anaphor in text.

The rest of this paper is organized as follows. In Section 2, we review related work on exploring syntactic structured information in pronoun resolution and their comparison with our method. Section 3 first presents a dynamic-expansion tree span scheme by automatically expanding the shortest path to include necessary structured information, such as predicate- and antecedent competitor-related information. Then it presents a context-sensitive convolution tree kernel, which not only enumerates context-free sub-trees but also context-sensitive sub-trees by considering their ancestor node paths as their contexts. Section 4 shows the experimental results. Finally, we conclude our work in Section 5.

2 Related Work

Related work on exploring syntactic structured information in pronoun resolution can be typically classified into three categories: parse tree-based search algorithms (Hobbs 1978), feature-based (Lappin and Leass 1994; Bergsma and Lin 2006) and tree kernel-based methods (Yang et al 2006).

As a representative for parse tree-based search algorithms, Hobbs (1978) found the antecedent for a given pronoun by searching the parse trees of current text. It processes one sentence at a time from current sentence to the first sentence in text until an antecedent is found. For each sentence, it searches the corresponding parse tree in a left-to-right breadth-first way. The first antecedent candidate, which satisfies hard constraints (such as gender and number agreement), would be returned as the antecedent. Since the search is completely done on the parse trees, one problem with the parse tree-based search algorithms is that the performance would heavily rely on the accuracy of the parse trees. Another problem is that such algorithms are not good enough to capture necessary structured information for pronoun resolution. There is still a big performance gap even on correct parse trees.

Similar to other NLP applications, feature-based methods have been widely applied in pronoun resolution to explore syntactic structured information from the parse trees. Lappin and Leass (1994) derived a set of salience measures (e.g. subject, object or accusative emphasis) with manually assigned weights from the syntactic structure output by McCord's Slot Grammar parser. The candidate with the highest salience score would be selected as the antecedent. Bergsma and Lin (2006) presented an approach to pronoun resolution based on syntactic paths. Through a simple bootstrapping procedure, highly co-reference paths can be learned reliably to handle previously challenging instances and robustly address traditional syntactic co-reference constraints. Although feature-based methods dominate on exploring syntactic structured information in the literature of pronoun resolution, there still exist two problems with them. One problem is that the structured features have to be selected and defined manually, usually by linguistic intuition. Another problem is that they may fail to effectively capture complex structured parse tree information.

As for tree kernel-based methods, Yang et al (2006) captured syntactic structured information for pronoun resolution by using the convolution tree kernel (Collins and Duffy 2001) to measure the common sub-trees enumerated from the parse trees and achieved quite success on the ACE 2003 corpus. They also explored different tree span schemes and found that the simple-expansion scheme performed best. One problem with their method is that the sub-trees enumerated in Collins and Duffy's kernel computation are context-free, that is, they do not consider the information outside the sub-trees. As a result, their ability of exploring syntactic structured information is much limited. Another problem is that, among the three explored schemes, there exists no obvious overwhelming one, which can well cover syntactic structured information.

The above discussion suggests that structured information in the parse trees may not be well utilized in the previous researches, regardless of feature-based or tree kernel-based methods. This paper follows tree kernel-based methods. Compared with Collins and Duffy's kernel and its application in pronoun resolution (Yang et al 2006), the context-sensitive convolution tree kernel enumerates not only context-free sub-trees but also context-sensitive sub-trees by taking their ancestor node paths into consideration. Moreover, this paper also implements a dynamic-expansion tree span scheme by taking predicate- and antecedent competitor-related information into consideration.

3 Context Sensitive Convolution Tree Kernel for Pronoun Resolution

In this section, we first propose an algorithm to dynamically determine a proper tree span for pronoun resolution and then present a context-sensitive convolution tree kernel to compute similarity between two tree spans. In this paper, all the texts are parsed using the Charniak parser (Charniak 2001) based on which the tree span is determined.

3.1 Dynamic-Expansion Tree Span Scheme

Normally, parsing is done on the sentence level. To deal with the cases that an anaphor and an antecedent candidate do not occur in the same sentence, we construct a pseudo parse tree for an entire text by attaching the parse trees of all its sentences to an upper "S" node, similar to Yang et al (2006).

Given the parse tree of a text, the problem is how to choose a proper tree span to well cover syntactic structured information in the tree kernel computation. Generally, the more a tree span includes, the more syntactic structured information would be provided, at the expense of more noisy information. Figure 2 shows the three tree span schemes explored in Yang et al (2006): Min-Expansion (only including the shortest path connecting the anaphor and the antecedent candidate), Simple-Expansion (containing not only all the nodes in Min-Expansion but also the first level children of these nodes) and Full-Expansion (covering the sub-tree between the anaphor and the candidate), such as the sub-trees inside the dash circles of Figures 2(a), 2(b) and 2(c) respectively. It is found (Yang et al 2006) that the simple-expansion tree span scheme performed best on the ACE 2003 corpus in pronoun resolution. This suggests that inclusion of more structured information in the tree span may not help in pronoun resolution.

To better capture structured information in the parse tree, this paper presents a dynamic-expansion scheme by trying to include necessary structured information in a parse tree. The intuition behind our scheme is that predicate- and antecedent competitor- (all the other compatible[1] antecedent candidates between the anaphor and the considered antecedent candidate) related information plays a critical role in pronoun resolution. Given an anaphor and an antecedent candidate, e.g. "Mary" and "her" as shown in Figure 1, this is done by:

1) Determining the min-expansion tree span via the shortest path, as shown in Figure 1(a).
2) Attaching all the antecedent competitors along the corresponding paths to the shortest path. As shown in Figure 1(b), "the woman" is attached while "the room" is not attached since the former is compatible with the anaphor and the latter is not compatible with the anaphor. In this way, the competition between the considered candidate and other compatible candidates can be included in the tree span. In some sense, this is a natural extension of the twin-candidate learning approach proposed in Yang et al (2003), which explicitly models the competition between two antecedent candidates.
3) For each node in the tree span, attaching the path from the node to the predicate terminal node if it is a predicate-headed node. As shown in Figure 1(c), "said" and "bit" are attached.
4) Pruning those nodes (except POS nodes) with the single in-arc and the single out-arc and with its syntactic phrase type same as its child node. As shown in Figure 1(d), the left child of the "SBAR" node, the "NP" node, is removed and the sub-tree (NP the/DT woman/NN) is attached to the "SBAR" node directly.

To show the difference among min-, simple-, full- and dynamic-expansion schemes, Figure 2 compares them for three different sentences, given the anaphor "her/herself" and the antecedent candidate "Mary". It shows that:

- Min-, simple- and full-expansion schemes have the same tree spans (except the word nodes) for the three sentences regardless of the difference among the sentences while the dynamic-expansion scheme can adapt to difference ones.
- Normally, the min-expansion scheme is too simple to cover necessary information (e.g. "the woman" in the 1st sentence is missing).
- The full-expansion scheme can cover all the information at the expense of much noise (e.g. "the man in that room" in the 2nd sentence).
- The simple-expansion scheme can cover some necessary predicate-related information (e.g. "said" and "bit" in the sentences). However, it may introduce some noise (e.g. the left child of

[1] With matched number, person and gender agreements.

the "SBAR" node, the "NP" node, may not be necessary in the 2nd sentence) and ignore necessary antecedent competitor-related information (e.g. "the woman" in the 1st sentence).

- The dynamic-expansion scheme normally works well. It can not only cover predicate-related information but also structured information related with the competitors of the considered antecedent candidate. In this way, the competition between the considered antecedent candidate and other compatible candidates can be included in the dynamic-expansion scheme.

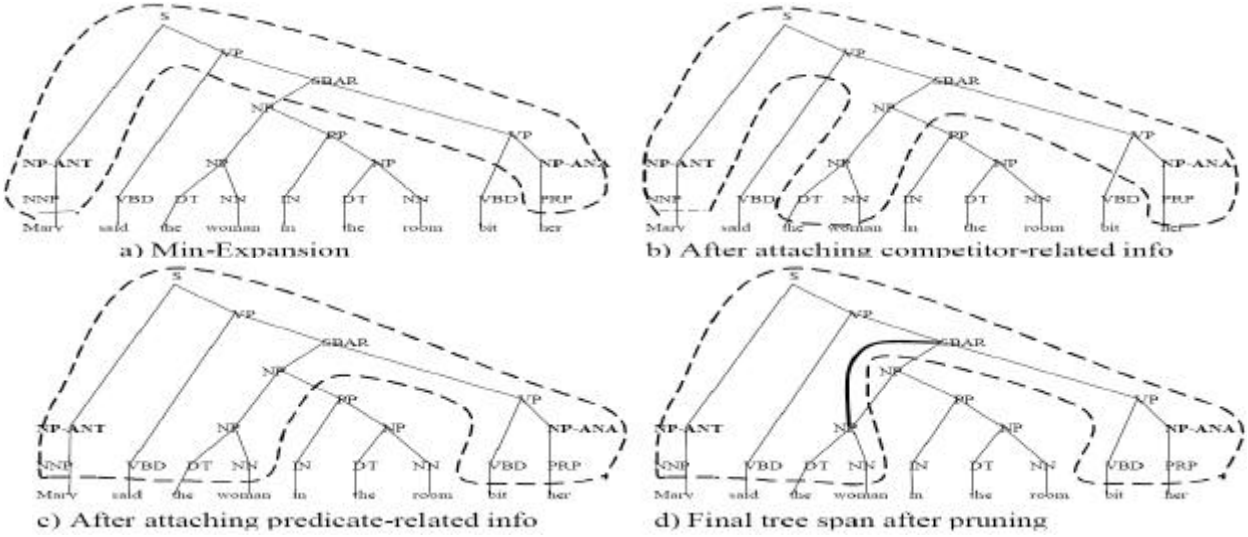


Figure 1: Dynamic-Expansion Tree Span Scheme

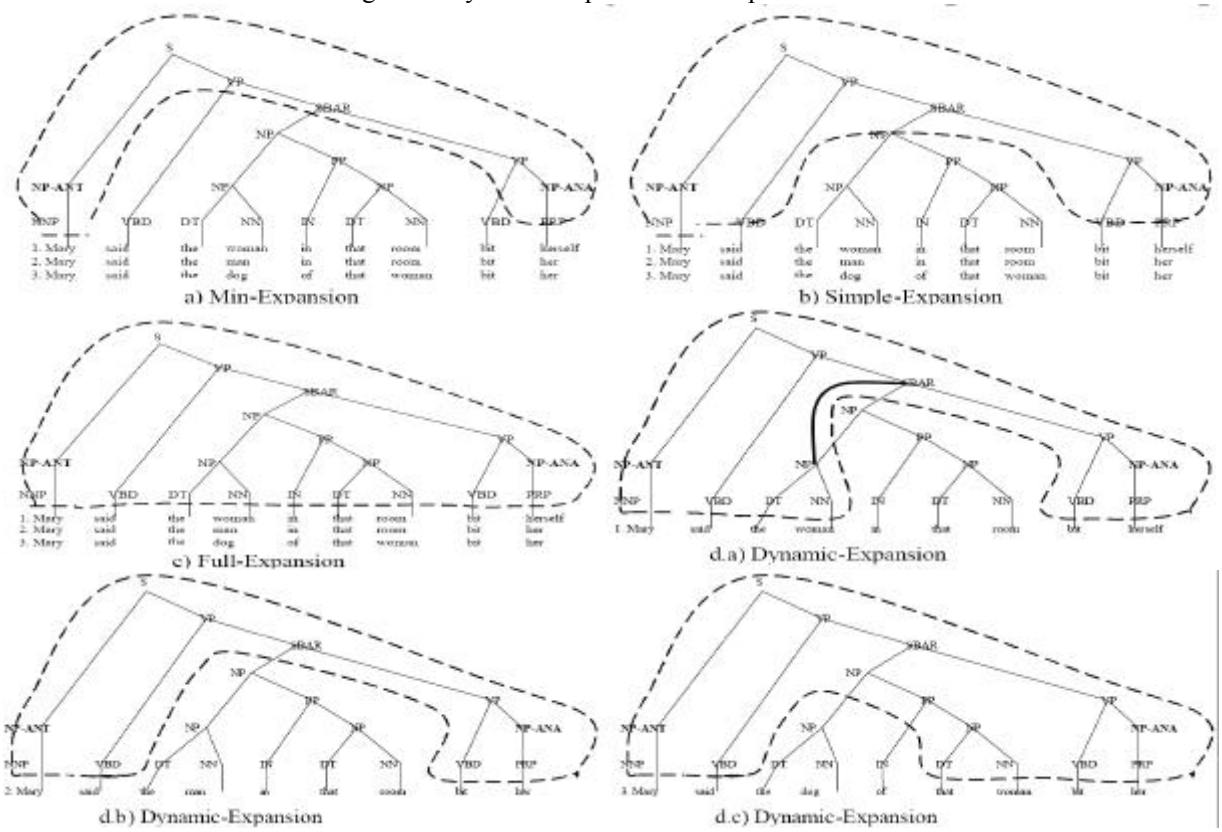


Figure 2: Comparison of Min-, Simple-, Full-and Dynamic-Expansions: More Examples

3.2 Context-Sensitive Convolution Tree Kernel

Given any tree span scheme, e.g. the dynamic-expansion scheme in the last subsection, we now study how to measure the similarity between two tree spans using a convolution tree kernel.

A convolution kernel (Haussler D., 1999) aims to capture structured information in terms of sub-structures. As a specialized convolution kernel, the convolution tree kernel, proposed in Collins and Duffy (2001), counts the number of common sub-trees (sub-structures) as the syntactic structure similarity between two parse trees. This convolution tree kernel has been successfully applied by Yang et al (2006) in pronoun resolution. However, there is one problem with this tree kernel: the sub-trees involved in the tree kernel computation are context-free (That is, they do not consider the information outside the sub-trees.). This is contrast to the tree kernel proposed in Culota and Sorensen (2004) which is context-sensitive, that is, it considers the path from the tree root node to the sub-tree root node. In order to integrate the advantages of both tree kernels and resolve the problem in Collins and Duffy's kernel, this paper applies the same context-sensitive convolution tree kernel, proposed by Zhou et al (2007) on relation extraction. It works by taking ancestral information (i.e. the root node path) of sub-trees into consideration:

$$K_C(T[1],T[2]) = \sum_{i=1}^{m} \sum_{\substack{n_1^i[1] \in N_1^i[1] \\ n_1^i[2] \in N_1^i[2]}} \Delta(n_1^i[1], n_1^i[2]) \quad (1)$$

where $N_1^i[j]$ is the set of root node paths with length i in tree $T[j]$ while the maximal length of a root node path is defined by m; and $\Delta(n_1^i[1], n_1^i[2])$ counts the common context-sensitive sub-trees rooted at root node paths $n_1^i[1]$ and $n_1^i[2]$. In the tree kernel, a sub-tree becomes context-sensitive via the "root node path" moving along the sub-tree root. For more details, please refer to Zhou et al (2007).

4 Experimentation

This paper focuses on the third-person pronoun resolution and, in all our experiments, uses the ACE 2003 corpus for evaluation. This ACE corpus contains ~3.9k pronouns in the training data and ~1.0k pronouns in the test data.

Similar to Soon et al (2001), an input raw text is first preprocessed automatically by a pipeline of NLP components, including sentence boundary detection, POS tagging, named entity recognition and phrase chunking, and then a training or test instance is formed by a pronoun and one of its antecedent candidates. During training, for each anaphor encountered, a positive instance is created by pairing the anaphor and its closest antecedent while a set of negative instances is formed by pairing the anaphor with each of the non-coreferential candidates. Based on the training instances, a binary classifier is generated using a particular learning algorithm. In this paper, we use SVMLight deleveloped by Joachims (1998). During resolution, an anaphor is first paired in turn with each preceding antecedent candidate to form a test instance, which is presented to a classifier. The classifier then returns a confidence value indicating the likelihood that the candidate is the antecedent. Finally, the candidate with the highest confidence value is selected as the antecedent. In this paper, the NPs occurring within the current and previous two sentences are taken as the initial antecedent candidates, and those with mismatched number, person and gender agreements are filtered out. On average, an anaphor has ~7 antecedent candidates. The performance is evaluated using F-measure instead of accuracy since evaluation is done on all the pronouns occurring in the data.

Scheme/*m*	1	2	3	4
Min	78.5	79.8	80.8	80.8
Simple	79.8	81.0	81.7	81.6
Full	78.3	80.1	81.0	81.1
Dynamic	80.8	82.3	83.0	82.9

Table 1: Comparison of different context-sensitive convolution tree kernels and tree span schemes (with entity type info attached at both the anaphor and the antecedent candidate nodes by default)

In this paper, the *m* parameter in our context-sensitive convolution tree kernel as shown in Equation (1) indicates the maximal length of root node paths and is optimized to 3 using 5-fold cross validation on the training data. Table 1 systematically evaluates the impact of different *m* in our context-sensitive convolution tree kernel and compares our dynamic-expansion tree span scheme with the existing three tree span schemes, min-,

simple- and full-expansions as described in Yang et al (2006). It also shows that that our tree kernel achieves best performance with m = 3 on the test data, which outperforms the one with m = 1 by ~2.2 in F-measure. This suggests that the parent and grandparent nodes of a sub-tree contain much information for pronoun resolution while considering more ancestral nodes doesnot further improve the performance. This may be due to that, although our experimentation on the training data indicates that more than 90% (on average) of subtrees has a root node path longer than 3 (since most of the subtrees are deep from the root node and more than 90% of the parsed trees are deeper than 6 levels in the ACE 2003 corpus), including a root node path longer than 3 may be vulnerable to the full parsing errors and have negative impact. It also shows that our dynamic-expansion tree span scheme outperforms min-expansion, simple-expansion and full-expansion schemes by ~2.4, ~1.2 and ~2.1 in F-measure respectively. This suggests the usefulness of dynamically expanding tree spans to cover necessary structured information in pronoun resolution. In all the following experiments, we will apply our tree kernel with m=3 and the dynamic-expansion tree span scheme by default, unless specified.

We also evaluate the contributions of antecedent competitor-related information, predicate-related information and pruning in our dynamic-expansion tree span scheme by excluding one of them from the dynamic-expansion scheme. Table 2 shows that 1) antecedent competitor-related information contributes much to our scheme; 2) predicate-related information contributes moderately; 3) pruning only has slight contribution. This suggests the importance of including the competition in the tree span and the effect of predicate-argument structures in pronoun resolution. This also suggests that our scheme can well make use of such predicate- and antecedent competitor-related information.

Dynamic Expansion	Effect
- Competitors-related Info	81.1(-1.9)
- Predicates-related Info	82.2 (-0.8)
- Pruning	82.8(-0.2)
All	83.0

Table 2: Contributions of different factors in our dynamic-expansion tree span scheme

Table 3 compares the performance of different tree span schemes for pronouns with antecedents in different sentences apart. It shows that our dynamic-expansion scheme is much more robust than other schemes with the increase of sentences apart.

Scheme / #Sentences Apart	0	1	2
Min	86.3	76.7	39.6
Simple	86.8	77.9	43.8
Full	86.6	77.4	35.4
Dynamic	87.6	78.8	54.2

Table 3: Comparison of tree span schemes with antecedents in different sentences apart

5 Conclusion

Syntactic structured information holds great potential in many NLP applications. The purpose of this paper is to well capture syntactic structured information in pronoun resolution. In this paper, we proposes a context-sensitive convolution tree kernel to resolve two critical problems in previous researches in pronoun resolution by first automatically determining a dynamic-expansion tree span, which effectively covers structured information in the parse trees by taking predicate- and antecedent competitor-related information into consideration, and then applying a context-sensitive convolution tree kernel, which enumerates both context-free sub-trees and context-sensitive sub-trees. Evaluation on the ACE 2003 corpus shows that our dynamic-expansion tree span scheme can better capture necessary structured information than the existing tree span schemes and our tree kernel can better model structured information than the state-of-the-art Collins and Duffy's kernel.

For the future work, we will focus on improving the context-sensitive convolution tree kernel by better modeling context-sensitive information and exploring new tree span schemes by better incorporating useful structured information. In the meanwhile, a more detailed quantitative evaluation and thorough qualitative error analysis will be performed to gain more insights.

Acknowledgement

This research is supported by Project 60673041 under the National Natural Science Foundation of China and Project 2006AA01Z147 under the "863" National High-Tech Research and Development of China.

References

Aone C and Bennett W.W. (1995). Evaluating automated and manual acquisition of anaphora resolution strategies. *ACL'1995*:122-129.

Bergsma S. and Lin D.K.(2006). Bootstrapping path-based pronoun resolution. *COLING-ACL'2006*: 33-40.

Charniak E. (2001). Immediate-head Parsing for Language Models. *ACL'2001*: 129-137. Toulouse, France

Collins M. (1999) Head-driven statistical models for natural language parsing. ***Ph.D. Thesis.*** University of Pennsylvania.

Collins M. and Duffy N. (2001). Convolution Kernels for Natural Language. *NIPS'2001*: 625-632. Cambridge, MA

Culotta A. and Sorensen J. (2004). Dependency tree kernels for relation extraction. *ACL'2004*. 423-429. 21-26 July 2004. Barcelona, Spain.

Haussler D. (1999). Convolution Kernels on Discrete Structures. *Technical Report UCS-CRL-99-10*, University of California, Santa Cruz.

Hobbs J. (1978). Resolving pronoun references. *Lingua*. 44:339-352.

Joachims T. (1998). Text Categorization with Support Vector Machine: learning with many relevant features. *ECML-1998*: 137-142. Chemnitz, Germany

Lappin S. and Leass H. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*. 20(4):526-561.

Mitkov R. (1998). Robust pronoun resolution with limited knowledge. *COLING-ACL'1998*:869-875. Montreal, Canada.

Moschitti A. (2004). A study on convolution kernels for shallow semantic parsing. *ACL'2004*:335-342.

Pradhan S., Hacioglu K., Krugler V., Ward W., Martin J.H. and Jurafsky D. (2005). Support Vector Learning for Semantic Argument Classification. *Machine Learning*. 60(1):11-39.

Soon W. Ng H.T.and Lim D. (2001). A machine learning approach to creference resolution of noun phrases. *Computational Linguistics*. 27(4): 521-544.

Yang X.F., Zhou G.D., Su J. and Tan C.L., Coreference Resolution Using Competition Learning Approach, *ACL'2003)*:176-183. Sapporo, Japan, 7-12 July 2003.

Yang X.F., Su J. and Tan C.L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. *COLING-ACL'2006*: 41-48.

Zelenko D., Aone C. and Richardella. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*. 3(Feb):1083-1106.

Zhang M., Zhang J., Su J. and Zhou G.D. (2006). A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features. *COLING-ACL-2006*: 825-832. Sydney, Australia

Zhou G.D., Su J. Zhang J. and Zhang M. (2005). Exploring various knowledge in relation extraction. *ACL'2005*. 427-434. 25-30 June, Ann Arbor, Michgan, USA.

Zhou G.D., Zhang M., Ji D.H. and Zhu Q.M. (2007). Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. EMNLP-CoNLL'2007

Semi-Supervised Learning for Relation Extraction

ZHOU GuoDong LI JunHui QIAN LongHua ZHU Qiaoming
Jiangsu Provincial Key Lab for Computer Information Processing Technology
School of Computer Science and Technology
Soochow Univ., Suzhou, China 215006
Email : {gdzhou, lijunhui, qianlonghua, qmzhu}@suda.edu.cn

Abstract

This paper proposes a semi-supervised learning method for relation extraction. Given a small amount of labeled data and a large amount of unlabeled data, it first bootstraps a moderate number of weighted support vectors via SVM through a co-training procedure with random feature projection and then applies a label propagation (LP) algorithm via the bootstrapped support vectors. Evaluation on the ACE RDC 2003 corpus shows that our method outperforms the normal LP algorithm via all the available labeled data without SVM bootstrapping. Moreover, our method can largely reduce the computational burden. This suggests that our proposed method can integrate the advantages of both SVM bootstrapping and label propagation.

1 Introduction

Relation extraction is to detect and classify various predefined semantic relations between two entities from text and can be very useful in many NLP applications such as question answering, e.g. to answer the query "Who is the president of the United States?", and information retrieval, e.g. to expand the query "George W. Bush" with "the president of the United States" via his relationship with "the United States".

During the last decade, many methods have been proposed in relation extraction, such as supervised learning (Miller et al 2000; Zelenko et al 2003; Culota and Sorensen 2004; Zhao and Grishman 2005; Zhang et al 2006; Zhou et al 2005, 2006), semi-supervised learning (Brin 1998; Agichtein and Gravano 2000; Zhang 2004; Chen et al 2006), and unsupervised learning (Hasegawa et al 2004; Zhang et al 2005). Among these methods, supervised learning-based methods perform much better than the other two alternatives. However, their performance much depends on the availability of a large amount of manually labeled data and it is normally difficult to adapt an existing system to other applications and domains. On the other hand, unsupervised learning-based methods do not need the definition of relation types and the availability of manually labeled data. However, they fail to classify exact relation types between two entities and their performance is normally very low. To achieve better portability and balance between human efforts and performance, semi-supervised learning has drawn more and more attention recently in relation extraction and other NLP applications.

This paper proposes a semi-supervised learning method for relation extraction. Given a small amount of labeled data and a large amount of unlabeled data, our proposed method first bootstraps a moderate number of weighted support vectors from all the available data via SVM using a co-training procedure with random feature projection and then applies a label propagation (LP) algorithm to capture the manifold structure in both the labeled and unlabeled data via the bootstrapped support vectors. Compared with previous methods, our method can integrate the advantages of both SVM bootstrapping in learning critical instances for the labeling function and label propagation in capturing the manifold structure in both the labeled and unlabeled data to smooth the labeling function.

The rest of this paper is as follows. In Section 2, we review related semi-supervised learning work in relation extraction. Then, the LP algorithm via bootstrapped support vectors is proposed in Section 3 while Section 4 shows the experimental results. Finally, we conclude our work in Section 5.

2 Related Work

Generally, supervised learning is preferable to unsupervised learning due to prior knowledge in the

annotated training data and better performance. However, the annotated data is usually expensive to obtain. Hence, there has been growing interest in semi-supervised learning, aiming at inducing classifiers by leveraging a small amount of labeled data and a large amount of unlabeled data. Related work in relation extraction using semi-supervised learning can be classified into two categories: bootstrapping-based (Brin 1998; Agichtein and Gravano 2000; Zhang 2004) and label propagation(LP)-based (Chen et al 2006).

Currently, bootstrapping-based methods dominate semi-supervised learning in relation extraction. Bootstrapping works by iteratively classifying unlabeled instances and adding confidently classified ones into labeled data using a model learned from augmented labeled data in previous iteration. Brin (1998) proposed a bootstrapping-based method on the top of a self-developed pattern matching-based classifier to exploit the duality between patterns and relations. Agichtein and Gravano (2000) shared much in common with Brin (1998). They employed an existing pattern matching-based classifier (i.e. SNoW) instead. Zhang (2004) approached the much simpler relation classification sub-task by bootstrapping on the top of SVM. Although bootstrapping-based methods have achieved certain success, one problem is that they may not be able to well capture the manifold structure among unlabeled data.

As an alternative to the bootstrapping-based methods, Chen et al (2006) employed a LP-based method in relation extraction. Compared with bootstrapping, the LP algorithm can effectively combine labeled data with unlabeled data in the learning process by exploiting the manifold structure (e.g. the natural clustering structure) in both the labeled and unlabeled data. The rationale behind this algorithm is that the instances in high-density areas tend to carry the same labels. The LP algorithm has also been successfully applied in other NLP applications, such as word sense disambiguation (Niu et al 2005), text classification (Szummer and Jaakkola 2001; Blum and Chawla 2001; Belkin and Niyogi 2002; Zhu and Ghahramani 2002; Zhu et al 2003; Blum et al 2004), and information retrieval (Yang et al 2006). However, one problem is its computational burden, especially when a large amount of labeled and unlabeled data is taken into consideration.

In order to take the advantages of both bootstrapping and label propagation, our proposed method propagates labels via bootstrapped support vectors. On the one hand, our method can well capture the manifold structure in both the labeled and unlabeled data. On the other hand, our method can largely reduce the computational burden in the normal LP algorithm via all the available data.

3 Label Propagation via Bootstrapped Support Vectors

The idea behind our LP algorithm via bootstrapped support vectors is that, instead of propagating labels through all the available labeled data, our method propagates labels through critical instances in both the labeled and unlabeled data. In this paper, we use SVM as the underlying classifier to bootstrap a moderate number of weighted support vectors for this purpose. This is based on an assumption that the manifold structure in both the labeled and unlabeled data can be well preserved through the critical instances (i.e. the weighted support vectors bootstrapped from all the available labeled and unlabeled data). The reason why we choose SVM is that it represents the state-of-the-art in machine learning research and there are good implementations of the algorithm available. In particular, SVMLight (Joachims 1998) is selected as our classifier. For efficiency, we apply the *one vs. others* strategy, which builds K classifiers so as to separate one class from all others. Another reason is that we can adopt the weighted support vectors returned by the bootstrapped SVMs as the critical instances, via which label propagation is done.

3.1 Bootstrapping Support Vectors

This paper modifies the SVM bootstrapping algorithm BootProject(Zhang 2004) to bootstrap support vectors. Given a small amount of labeled data and a large amount of unlabeled data, the modified BootProject algorithm bootstraps on the top of SVM by iteratively classifying unlabeled instances and moving confidently classified ones into labeled data using a model learned from the augmented labeled data in previous iteration, until not enough unlabeled instances can be classified confidently. Figure 1 shows the modified BootProject algorithm for bootstrapping support vectors.

Assume:
L : the labeled data;
U : the unlabeled data;
S : the batch size (100 in our experiments);
P : the number of views(feature projections);
r : the number of classes (including all the relation (sub)types and the non-relation)

BEGIN

REPEAT
FOR i = 1 to P DO
Generate projected feature space F_i from the original feature space F ;
Project both L and U onto F_i, thus generate L_i and U_i ;
Train SVM classifier SVM_{ij} on L_i for each class $r_j (j = 1 \ldots r)$;
Run SVM_{ij} on U_i for each class $r_j (j = 1 \ldots r)$
END FOR
Find (at most) S instances in U with the highest agreement (with threshold 70% in our experiments) and the highest average SVM-returned confidence value (with threshold 1.0 in our experiments);
Move them from U to L;
UNTIL not enough unlabeled instances (less than 10 in our experiments) can be confidently classified;
Return all the (positive and negative) support vectors included in all the latest SVM classifiers SVM_{ij} with their collective weight (absolute alpha*y) information as the set of bootstrapped support vectors to act as the labeled data in the LP algorithm;
Return U (those hard cases which can not be confidently classified) to act as the unlabeled data in the LP algorithm;

END

Figure 1: The algorithm for bootstrapping support vectors

In particular, this algorithm generates multiple overlapping "views" by projecting from the original feature space. In this paper, feature views with random feature projection, as proposed in Zhang (2004), are explored. Section 4 will discuss this issue in more details. During the iterative training process, classifiers trained on the augmented labeled data using the projected views are then asked to vote on the remaining unlabeled instances and those with the highest probability of being correctly labeled are chosen to augment the labeled data.

During the bootstrapping process, the support vectors included in all the trained SVM classifiers (for all the relation (sub)types and the non-relation) are bootstrapped (i.e. updated) at each iteration. When the bootstrapping process stops, all the (positive and negative) support vectors included in the SVM classifiers are returned as bootstrapped support vectors with their collective weights (absolute a*y) to act as the labeled data in the LP algorithm and all the remaining unlabeled instances (i.e. those hard cases which can not be confidently classified in the bootstrapping process) in the unlabeled data are returned to act as the unlabeled data in the LP algorithm. Through SVM bootstrapping, our LP algorithm will only depend on the critical instances (i.e. support vectors with their weight information bootstrapped from all the available labeled and unlabeled data) and those hard instances, instead of all the available labeled and unlabeled data.

3.2 Label Propagation

In the LP algorithm (Zhu and Ghahramani 2002), the manifold structure in data is represented as a connected graph. Given the labeled data (the above bootstrapped support vectors with their weights) and unlabeled data (the remaining hard instances in the unlabeled data after bootstrapping, including all the test instances for evaluation), the LP algorithm first represents labeled and unlabeled instances as vertices in a connected graph, then propagates the label information from any vertex to nearby vertex through weighted edges and finally infers the labels of unlabeled instances until a global stable stage is achieved. Figure 2 presents the label propagation algorithm on bootstrapped support vectors in details.

Assume:

Y : the $n*r$ labeling matrix, where y_{ij} represents the probability of vertex $x_i (i = 1 \ldots n)$ with label $r_j (j = 1 \ldots r)$ (including the non-relation label);

Y_L : the top l rows of Y^0. Y_L corresponds to the l labeled instances;

Y_U : the bottom u rows of Y^0. Y_U corresponds to the u unlabeled instances;

$\overline{T}$: a $n*n$ matrix, with $\bar{t}_{ij}$ is the probability jumping from vertex x_i to vertex x_j;

BEGIN (the algorithm)

Initialization:

1) Set the iteration index $t = 0$;

2) Let Y^0 be the initial soft labels attached to each vertex;

3) Let Y_L^0 be consistent with the labeling in the labeled (including all the relation (sub)types and the non-relation) data, where y_{ij}^0 = the weight of the bootstrapped support vector if x_i has label r_j (Please note that r_j can be the non-relation label) and 0 otherwise;

4) Initialize Y_U^0;

REPEAT

Propagate the labels of any vertex to nearby vertices by $Y^{t+1} = \overline{T} Y^t$;

Clamp the labeled data, that is, replace Y_L^{t+1} with Y_L^0;

UNTIL Y converges(e.g. Y_L^{t+1} converges to Y_L^0);

Assign each unlabeled instance with a label: for $x_i (l \prec i \le n)$, find its label with $\arg\max_j y_{ij}$;

END (the algorithm)

Figure 2: The LP algorithm

Here, each vertex corresponds to an instance, and the edge between any two instances x_i and x_j is weighted by w_{ij} to measure their similarity. In principle, larger edge weights allow labels to travel through easier. Thus the closer the instances are, the more likely they have similar labels. The algorithm first calculates the weight w_{ij} using a kernel, then transforms it to $t_{ij} = p(j \to i) = w_{ij} / \sum_{k=1}^{n} w_{kj}$, which measures the probability of propagating a label from instance x_j to instance x_i, and finally normalizes t_{ij} row by row using $\bar{t}_{ij} = t_{ij} / \sum_{k=1}^{n} t_{ik}$ to maintain the class probability interpretation of the labeling matrix Y.

During the label propagation process, the label distribution of the labeled data is clamped in each loop using the weights of the bootstrapped support vectors and acts like forces to push out labels through the unlabeled data. With this push originates from the labeled data, the label boundaries will be pushed much faster along edges with larger weights and settle in gaps along those with lower weights. Ideally, we can expect that w_{ij} across different classes should be as small as possible and w_{ij} within the same class as big as possible. In this way, label propagation happens within the same class most likely.

This algorithm has been shown to converge to a unique solution (Zhu and Ghahramani 2002), which can be obtained without iteration in theory, and the initialization of Y_U^0 (the unlabeled data) is not important since Y_U^0 does not affect its estimation. However, proper initialization of Y_U^0 actually helps the algorithm converge more rapidly in practice. In this paper, each row in Y_U^0 is initialized to the average similarity with the labeled instances.

4 Experimentation

This paper uses the ACE RDC 2003 corpus provided by LDC for evaluation. This corpus is gathered from various newspapers, newswires and broadcasts.

Method	LP via bootstrapped (weighted) SVs	LP via bootstrapped (un-weighted) SVs	LP w/o SVM bootstrapping	SVM	(BootProject) SVM Bootstrapping
5%	46.5 (+1.4)	44.5 (+1.7)	43.1 (+1.0)	35.4 (-)	40.6 (+0.9)
10%	48.6 (+1.7)	46.5 (+2.1)	45.2 (+1.5)	38.6 (-)	43.1 (+1.4)
25%	51.7 (+1.9)	50.4 (+2.3)	49.6 (+1.8)	43.9 (-)	47.8 (+1.7)
50%	53.6 (+1.8)	52.6 (+2.2)	52.1 (+1.7)	47.2 (-)	50.5 (+1.6)
75%	55.2 (+1.3)	54.5 (+1.8)	54.2 (+1.2)	53.1 (-)	53.9 (+1.2)
100%	56.2 (+1.0)	55.8 (+1.3)	55.6 (+0.8)	55.5 (-)	55.8 (+0.7)

Table 1: Comparison of different methods using a state-of-the-art linear kernel on the ACE RDC 2003 corpus (The numbers inside the parentheses indicate the increases in F-measure if we add the ACE RDC 2004 corpus as the unlabeled data)

4.1 Experimental Setting

In the ACE RDC 2003 corpus, the training data consists of 674 annotated text documents (~300k words) and 9683 instances of relations. During development, 155 of 674 documents in the training set are set aside for fine-tuning. The test set is held out only for final evaluation. It consists of 97 documents (~50k words) and 1386 instances of relations. The ACE RDC 2003 task defines 5 relation types and 24 subtypes between 5 entity types, i.e. person, organization, location, facility and GPE. All the evaluations are measured on the 24 subtypes including relation identification and classification.

In all our experiments, we iterate over all pairs of entity mentions occurring in the same sentence to generate potential relation instances[1]. For better evaluation, we have adopted a state-of-the-art linear kernel as similarity measurements. In our linear kernel, we apply the same feature set as described in a state-of-the-art feature-based system (Zhou et al 2005): word, entity type, mention level, overlap, base phrase chunking, dependency tree, parse tree and semantic information. Given above various lexical, syntactic and semantic features, multiple overlapping feature views are generated in the bootstrapping process using random feature projection (Zhang 2004). For each feature projection in bootstrapping support vectors, a feature is randomly selected with probability p and therefore the eventually projected feature space has $p*F$ features on average, where F is the size of the original feature space. In this paper, p and the number of different views are fine-tuned to 0.5 and 10 [2] respectively using 5-fold cross validation on the training data of the ACE RDC 2003 corpus.

4.2 Experimental Results

Table 1 presents the F-measures[3] (the numbers outside the parentheses) of our algorithm using the state-of-the-art linear kernel on different sizes of the ACE RDC training data with all the remaining training data and the test data[4] as the unlabeled data on the ACE RDC 2003 corpus. In this paper, we only report the performance (averaged over 5 trials) with the percentages of 5%, 10%, 25%, 50%, 75% and 100%[5]. For example, our LP algorithm via bootstrapped (weighted) support vectors achieves the F-measure of 46.5 if using only 5% of the ACE RDC 2003 training data as the labeled data and the remaining training data and the test data in this corpus as the unlabeled data. Table 1

[1] In this paper, we only measure the performance of relation extraction on "true" mentions with "true" chaining of co-reference (i.e. as annotated by the corpus annotators) in the ACE corpora. We also explicitly model the argument order of the two mentions involved and only model explicit relations because of poor inter-annotator agreement in the annotation of implicit relations and their limited number.

[2] This suggests that the modified BootProject algorithm in the bootstrapping phase outperforms the SelfBoot algorithm (with p=1.0 and m=1) which uses all the features as the only view. In the related NLP literature, co-training has also shown to typically outperform self-bootstrapping.

[3] Our experimentation also shows that most of performance improvement with either bootstrapping or label propagation comes from gain in recall. Due to space limitation, this paper only reports the overall F-measure.

[4] In our label propagation algorithm via bootstrapped support vectors, the test data is only included in the second phase (i.e. the label propagation phase) and not used in the first phase (i.e. bootstrapping support vectors). This is to fairly compare different semi-supervised learning methods.

[5] We have tried less percentage than 5%. However, our experiments show that using much less data will suffer from performance un-stability. Therefore, we only report the performance with percentage not less than 5%.

also compares our method with SVM and the original SVM bootstrapping algorithm BootProject(i.e. bootstrapping on the top of SVM with feature projection, as proposed in Zhang (2004)). Finally, Table 1 compares our LP algorithm via bootstrapped (weighted by default) support vectors with other possibilities, such as the scheme via bootstrapped (un-weighted, i.e. the importance of support vectors is not differentiated) support vectors and the scheme via all the available labeled data (i.e. without SVM bootstrapping). Table 1 shows that:

1) Inclusion of unlabeled data using semi-supervised learning, including the SVM bootstrapping algorithm BootProject, the normal LP algorithm via all the available labeled and unlabeled data without SVM bootstrapping, and our LP algorithms via bootstrapped (either weighted or un-weighted) support vectors, consistently improves the performance, although semi-supervised learning has shown to typically decrease the performance when a lot of (enough) labeled data is available (Nigam 2001). This may be due to the insufficiency of labeled data in the ACE RDC 2003 corpus. Actually, most of relation subtypes in the two corpora much suffer from the data sparseness problem (Zhou et al 2006).
2) All the three LP algorithms outperform the state-of-the-art SVM classifier and the SVM bootstrapping algorithm BootProject. Especially, when a small amount of labeled data is available, the performance improvements by the LP algorithms are significant. This indicates the usefulness of the manifold structure in both labeled and unlabeled data and the powerfulness of the LP algorithm in modeling such information.
3) Our LP algorithms via bootstrapped (either weighted or un-weighted) support vectors outperforms the normal LP algorithm via all the available labeled data w/o SVM bootstrapping. For example, our LP algorithm via bootstrapped (weighted) support vectors outperforms the normal LP algorithm from 0.6 to 3.4 in F-measure on the ACE RDC 2003 corpus respectively when the labeled data ranges from 100% to 5%. This suggests that the manifold structure in both the labeled and unlabeled data can be well preserved via bootstrapped support vectors, especially when only a small amount of labeled data is available. This implies that weighted support vectors may represent the manifold structure (e.g. the decision boundary from where label propagation is done) better than the full set of data – an interesting result worthy more quantitative and qualitative justification in the future work.
4) Our LP algorithms via bootstrapped (weighted) support vectors perform better than LP algorithms via bootstrapped (un-weighted) support vectors by ~1.0 in F-measure on average. This suggests that bootstrapped support vectors with their weights can better represent the manifold structure in all the available labeled and unlabeled data than bootstrapped support vectors without their weights.
5) Comparison of SVM, SVM bootstrapping and label propagation with bootstrapped (weighted) support vectors shows that both bootstrapping and label propagation contribute much to the performance improvement.

Table 1 also shows the increases in F-measure (the numbers inside the parentheses) if we add all the instances in the ACE RDC 2004[6] corpus into the ACE RDC 2003 corpus in consideration as unlabeled data in all the four semi-supervised learning methods. It shows that adding more unlabeled data can consistently improve the performance. For example, compared with using only 5% of the ACE RDC 2003 training data as the labeled data and the remaining training data and the test data in this corpus as the unlabeled data, including the ACE RDC 2004 corpus as the unlabeled data increases the F-measures of 1.4 and 1.0 in our LP algorithm and the normal LP algorithm respectively. Table 1 shows that the contribution grows first when the labeled data begins to increase and reaches a maximum of ~2.0 in F-measure at a certain point.

Finally, it is found in our experiments that critical and hard instances normally occupy only 15~20% (~18% on average) of all the available labeled and unlabeled data. This suggests that, through bootstrapped support vectors, our LP algo-

[6] Compared with the ACE RDC 2003 task, the ACE RDC 2004 task defines two more entity types, i.e. weapon and vehicle, much more entity subtypes, and different 7 relation types and 23 subtypes between 7 entity types. The ACE RDC 2004 corpus from LDC contains 451 documents and 5702 relation instances.

rithm can largely reduce the computational burden since it only depends on the critical instances (i.e. bootstrapped support vectors with their weights) and those hard instances.

5 Conclusion

This paper proposes a new effective and efficient semi-supervised learning method in relation extraction. First, a moderate number of weighted support vectors are bootstrapped from all the available labeled and unlabeled data via SVM through a co-training procedure with feature projection. Here, a random feature projection technique is used to generate multiple overlapping feature views in bootstrapping using a state-of-the-art linear kernel. Then, a LP algorithm is applied to propagate labels via the bootstrapped support vectors, which, together with those hard unlabeled instances and the test instances, are represented as vertices in a connected graph. During the classification process, the label information is propagated from any vertex to nearby vertex through weighted edges and finally the labels of unlabeled instances are inferred until a global stable stage is achieved. In this way, the manifold structure in both the labeled and unlabeled data can be well captured by label propagation via bootstrapped support vectors. Evaluation on the ACE RDC 2004 corpus suggests that our LP algorithm via bootstrapped support vectors can take the advantages of both SVM bootstrapping and label propagation.

For the future work, we will systematically evaluate our proposed method on more corpora and explore better metrics of measuring the similarity between two instances.

Acknowledgement

This research is supported by Project 60673041 under the National Natural Science Foundation of China and Project 2006AA01Z147 under the "863" National High-Tech Research and Development of China.

References

ACE. (2000-2005). Automatic Content Extraction. http://www.ldc.upenn.edu/Projects/ACE/

Agichtein E. and Gravano L. (2000). Snowball: Extracting relations from large plain-text collections. Proceedings of the 5th ACM International Conference on Digital Libraries (ACMDL'2000).

Belkin, M. and Niyogi, P. (2002). Using Manifold Structure for Partially Labeled Classification. *NIPS 15*.

Blum A. and Chawla S. (2001). Learning from labeled and unlabeled data using graph mincuts. *ICML'2001*.

Blum A., Lafferty J., Rwebangira R and Reddy R. (2004). Semi-supervised learning using randomized mincuts. *ICML'2004*.

Brin S. (1998). Extracting patterns and relations from world wide web. *Proceedings of WebDB Workshop at 6th International Conference on Extending Database Technology*:172-183.

Charniak E. (2001). Immediate-head Parsing for Language Models. *ACL'2001*: 129-137. Toulouse, France

Chen J.X., Ji D.H., Tan C.L. and Niu Z.Y. (2006). Relation extraction using label propagation based semi-supervised learning. *COLING-ACL'2006*: 129-136. July 2006. Sydney, Australia.

Culotta A. and Sorensen J. (2004). Dependency tree kernels for relation extraction. *ACL'2004*. 423-429. 21-26 July 2004. Barcelona, Spain.

Hasegawa T., Sekine S. and Grishman R. (2004). Discovering relations among named entities form large corpora. *ACL'2004*. Barcelona, Spain.

Miller S., Fox H., Ramshaw L. and Weischedel R. (2000). A novel use of statistical parsing to extract information from text. *ANLP'2000*. 226-233. 29 April - 4 May 2000, Seattle, USA

Moschitti A. (2004). A study on convolution kernels for shallow semantic parsing. *ACL'2004*:335-342.

Nigam K.P. (2001). Using unlabeled data to improve text classification. Technical Report CMU-CS-01-126.

Niu Z.Y., Ji D.H., and Tan C.L. (2005). Word Sense Disambiguation Using Label Propagation Based Semi-supervised Learning. ACL'2005:395-402., Ann Arbor, Michigan, USA.

Szummer, M., & Jaakkola, T. (2001). Partially Labeled Classification with Markov Random Walks. *NIPS 14*.

Yang L.P., Ji D.H., Zhou G.D. and Nie Y. (2006). Document Re-ranking using cluster validation and label propagation. *CIKM'2006*. 5-11 Nov 2006. Arlington, Virginia, USA.

Zelenko D., Aone C. and Richardella. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*. 3(Feb):1083-1106.

Zhang M., Su J., Wang D.M., Zhou G.D. and Tan C.L. (2005). Discovering Relations from a Large Raw Corpus Using Tree Similarity-based Clustering, *IJCNLP'2005, Lecture Notes in Artificial Intelligence (LNAI 3651)*. 378-389.

Zhang M., Zhang J., Su J. and Zhou G.D. (2006). A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features. *COLING-ACL-2006*: 825-832. Sydney, Australia

Zhang Z. (2004). Weakly supervised relation classification for information extraction. CIKM'2004. 8-13 Nov 2004. Washington D.C. USA.

Zhao S.B. and Grishman R. (2005). Extracting relations with integrated information using kernel methods. *ACL'2005*: 419-426. Univ of Michigan-Ann Arbor，USA，25-30 June 2005.

Zhou G.D., Su J. Zhang J. and Zhang M. (2005). Exploring various knowledge in relation extraction. *ACL'2005*. 427-434. 25-30 June, Ann Arbor, Michgan, USA.

Zhou G.D., Su J. and Zhang M. (2006). Modeling commonality among related classes in relation extraction, *COLING-ACL'2006*: 121-128. Sydney, Australia.

Zhu, X. and Ghahramani, Z. (2002). Learning from Labeled and Unlabeled Data with Label Propagation. *CMU CALD Technical Report.* CMU-CALD-02-107.

Zhu, X., Ghahramani, Z. and Lafferty, J. (2003). Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. *ICML'2003*.

Story Link Detection based on Dynamic Information Extending

Xiaoyan Zhang Ting Wang Huowang Chen
Department of Computer Science and Technology, School of Computer,
National University of Defense Technology
No.137, Yanwachi Street, Changsha, Hunan 410073, P.R.China
{zhangxiaoyan, tingwang, hwchen}@nudt.edu.cn

Abstract

Topic Detection and Tracking refers to automatic techniques for locating topically related materials in streams of data. As the core technology of it, story link detection is to determine whether two stories are about the same topic. To overcome the limitation of the story length and the topic dynamic evolution problem in data streams, this paper presents a method of applying dynamic information extending to improve the performance of link detection. The proposed method uses previous latest related story to extend current processing story, generates new dynamic models for computing the similarity between the current two stories. The work is evaluated on the TDT4 Chinese corpus, and the experimental results indicate that story link detection using this method can make much better performance on all evaluation metrics.

1 Introduction

Topic Detection and Tracking (TDT) (Allan, 2002) refers to a variety of automatic techniques for discovering and threading together topically related material in streams of data such as newswire or broadcast news. Such automatic discovering and threading could be quite valuable in many applications where people need timely and efficient access to large quantities of information. Supported by such technology, users could be alerted with new events and new information about known events. By examining one or two stories, users define the topic described in them. Then with TDT technologies they could go to a large archive, find all the stories about this topic, and learn how it evolved.

Story link detection, as the core technology defined in TDT, is a task of determining whether two stories are about the same topic, or topically linked. In TDT, a topic is defined as "something that happens at some specific time and place" (Allan, 2002). Link detection is considered as the basis of other event-based TDT tasks, such as topic tracking, topic detection, and first story detection. Since story link detection focuses on the streams of news stories, it has its specific characteristic compared with the traditional Information Retrieval (IR) or Text Classification task: new topics usually come forth frequently during the procedure of the task, but nothing about them is known in advance.

The paper is organized as follows: Section 2 describes the procedure of story link detection; Section 3 introduces the related work in story link detection; Section 4 explains a baseline method which will be compared with the proposed dynamic method in Section 5; the experiment results and analysis are given in Section 6; finally, Section 7 concludes the paper.

2 Problem Definition

In the task definition of story link detection (NIST, 2003), a link detection system is given a sequence of time-ordered news source files $S = \langle S_1, S_2, S_3, \ldots, S_n \rangle$ where each S_i includes a set of stories, and a sequence of time-ordered story pairs $P = \langle P_1, P_2, P_3, \ldots, P_m \rangle$ where $P_i =$

$(s_{i1}, s_{i2}), s_{i1} \in S_j, s_{i2} \in S_k, 1 \leq i \leq m, 1 \leq j \leq k \leq n$. The system is required to make decisions on all story pairs to judge if they describe a same topic.

We formalize the procedure for processing a pair of stories as follows:

For a story pair $P_i = (s_{i1}, s_{i2})$:

1. Get background corpus B_i of P_i. According to the supposed application situation and the custom that people usually look ahead when they browse something, in TDT research the system is usually allowed to look ahead N (usually 10) source files when deciding whether the current pair is linked. So $B_i = \{S_1, S_2, S_3, \ldots, S_l\}$, where
$$l = \begin{cases} k+10 & , s_{i2} \in S_k \text{ and } (k+10) \leq n \\ n & , s_{i2} \in S_k \text{ and } (k+10) > n \end{cases}.$$

2. Produce the representation models (M_{i1}, M_{i2}) for two stories in P_i. $M = \{(f_s, w_s) \mid s \geq 1\}$, where f_s is a feature extracted from a story and w_s is the weight of the feature in the story. They are computed with some parameters counted from current story and the background.

3. Choose a similarity function F and computing the similarity between two models. If t is a predefined threshold and $F(M_{i1}, M_{i2}) \geq t$, then stories in P_i are topically linked.

3 Related Work

A number of works has been developed on story link detection. It can be classified into two categories: vector-based methods and probabilistic-based methods.

The vector space model is widely used in IR and Text Classification research. Cosine similarity between document vectors with $tf*idf$ term weighting (Connell et al., 2004) (Chen et al., 2004) (Allan et al., 2003) is also one of the best technologies for link detection. We have examined a number of similarity measures in story link detection, including cosine, Hellinger and Tanimoto, and found that cosine similarity produced outstanding results. Furthermore, (Allan et al., 2000) also confirms this conclusion among cosine, weighted sum, language modeling and Kullback-Leibler divergence in its story link detection research.

Probabilistic-based method has been proven to be very effective in several IR applications. One of its attractive features is that it is firmly rooted in the theory of probability, thereby allowing the researcher to explore more sophisticated models guided by the theoretical framework. (Nallapati and Allan, 2002) (Lavrenko et al., 2002) (Nallapati, 2003) all apply probability models (language model or relevance model) for story link detection. And the experiment results indicate that the performances are comparable with those using traditional vector space models, if not better.

On the basis of vector-based methods, this paper represents a method of dynamic information extending to improve the performance of story link detection. It makes use of the previous latest topically related story to extend the vector model of current being processed story. New dynamic models are generated for computing the similarity between two stories in current pair. This method resolves the problems of information shortage in stories and topic dynamic evolution in streams of data.

Before introducing the proposed method, we first describe a method which is implemented with vector model and cosine similarity function. This straight and classic method is used as a baseline to be compared with the proposed method.

4 Baseline Story Link Detection

The related work in story link detection shows that vector representation model with cosine function can be used to build the state-of-the-art story link detection systems. Many research organizations take this as their baseline system (Connell et al., 2004) (Yang et al., 2002). In this paper, we make a similar choice.

The baseline method represents each story as a vector in term space, where the coordinates represent the weights of the term features in the story. Each vector terms (or feature) is a single word plus its tag which is produced by a segmenter and part of speech tagger for Chinese. So if two tokens with same spelling are tagged with different tags, they will be taken as different terms (or features). It is notable that in it is independent between processing any two comparisons the baseline method.

4.1 Preprocessing

A preprocessing has been performed for TDT Chinese corpus. For each story we tokenize the text, tag the generated tokens, remove stop words, and then get a candidate set of terms for its vector model. After that, the term-frequency for each token in the story and the length of the story will also be acquired. In the baseline and dynamic methods, both training and test data are preprocessed in this way.

The segmenter and tagger used here is ICTCLAS [1] . The stop word list is composed of 507 terms. Although the term feature in the vector representation is the word plus its corresponding tag, we will ignore the tag information when filtering stop words, because almost all the words in the list should be filtered out whichever part of speech is used to tag them.

4.2 Feature Weighting

One important issue in the vector model is weighting the individual terms (features) that occur in the vector. Most IR systems employed the traditional $tf * idf$ weighting, which also provide the base for the baseline and dynamic methods in this paper. Furthermore, this paper adopts a dynamic way to compute the $tf * idf$ weighting:

$$w_i(f_i, d) = tf(f_i, d) * idf(f_i)$$

$$tf = t/(t + 0.5 + 1.5dl/dl_{avg})$$

$$idf = log((N + 0.5)/df)/log(N + 1)$$

where t is the term frequency in a story, dl is the length of a story, dl_{avg} is the average length of stories in the background corpus, N is the number of stories in the corpus, df is the number of the stories containing the term in the corpus.

The tf shows how much a term represents the story, while the idf reflects the distinctive ability of distinguishing current story from others. The dynamic attribute of the $tf * idf$ weighting lies in the dynamic computation of dl_{avg}, N and df. The background corpus used for statistics is incremental. As more story pairs are processed, more source files could be seen, and the background is expanding as well. Whenever the size of the background has changed, the values of dl_{avg}, N and df will update accordingly. We call this as incremental $tf*idf$ weighting. A story might have different term vectors in different story pairs.

[1] http://sewm.pku.edu.cn/QA/reference/ICTCLAS/FreeICTCLAS/

4.3 Similarity Function

Another important issue in the vector model is determining the right function to measure the similarity between two vectors. We have firstly tried three functions: cosine, Hellinger and Tanimoto, among which cosine function performs best for its substantial advantages and the most stable performance. So we consider the cosine function in baseline method.

Cosine similarity, as a classic measure and consistent with the vector representation, is simply an inner product of two vectors where each vector is normalized to the unit length. It represents cosine of the angle between two vector models $M_1 = \{(f_{1i}, w_{1i}), i \geq 1\}$ and $M_2 = \{(f_{2i}, w_{2i}), i \geq 1\}$.

$$\cos(M_1, M_2) = (\Sigma(w_{1i} \times w_{2i}))/\sqrt{(\Sigma w_{1i}^2)(\Sigma w_{2i}^2)}$$

Cosine similarity tends to perform best at full dimensionality, as in the case of comparing two stories. Performance degrades as one of the vectors becomes shorter. Because of the built-in length normalization, cosine similarity is less dependent on specific term weighting.

5 Dynamic Story Link Detection

5.1 Motivation

Investigation on the TDT corpus shows that news stories are usually short, which makes that their representation models are too sparse to reflect topics described in them. A possible method of solving this problem is to extend stories with other related information. The information can be synonym in a dictionary, related documents in external corpora, etc. However, extending with synonym is mainly adding repetitious information, which can not define the topics more clearly. On the other hand, topic-based research should be real-sensitive. The corpora in the same period as the test corpora are not easy to gather, and the number of related documents in previous period is very few. So it is also not feasible to extend the stories with related documents in other corpora. We believe that it is more reasonable that the best extending information may be the

story corpus itself. Following the TDT evaluation requirement, we will not use entire corpus at a time. Instead, when we process current pair of stories, we utilize all the stories before the current pair in the story corpus.

In addition, topics described by stories usually evolve along with time. A topic usually begins with a seminal event. After that, it will focus mainly on the consequence of the event or other directly related events as the time goes. When the focus in later stories has changed, the words used in them may change remarkably. Keeping topic descriptions unchanged from the beginning to the end is obviously improper. So topic representation models should also be updated as the topic emphases in stories has changed. Formerly we have planed to use related information to extend a story to make up the information shortage in stories. Considering more about topic evolution, we extend a story with its latest related story. In addition, up to now almost all research in story link detection takes the hypothesis that whether two stories in one pair are topically linked is independent of that in another pair. But we realize that if two stories in a pair describe a same topic, one story can be taken as related information to extend another story in later pairs. Compared with extending with more than one story, extending only with its latest related story can keep representation of the topic as fresh as possible, and avoid extending too much similar information at the same time, which makes the length of the extended vector too long. Since the vector will be renormalized, a too big length means evidently decreasing the weight of an individual feature which will instead cause a lower cosine similarity. This idea has also been confirmed by the experiment showing that the performance extending with one latest related story is superior to that extending with more than one related story, as described in section 6.3. The experiment results also show that this method of dynamic information extending apparently improves the performance of story link detection.

5.2 Method Description

The proposed dynamic method is actually the baseline method plus dynamic information extending. The preprocessing, feature weighting and similarity computation in dynamic method are similar as those in baseline method. However, the vector representation for a story here is dynamic. This method needs a training corpus to get the *extending threshold* deciding whether a story should be used to extend another story in a pair. We split the sequence of time-ordered story pairs into two parts: the former is for training and the later is for testing. The following is the processing steps:

1. Preprocess to create a set of terms for representing each story as a term vector, which is same as baseline method.

2. Run baseline system on the training corpora and find an optimum *topically link threshold*. We take this threshold as *extending threshold*. The *topically link threshold* used for making link decision in dynamic method is another predefined one.

3. Along with the ordered story pairs in the test corpora, repeat a) and b):

   (a) When processing a pair of stories $P_i = (s_{i1}, s_{i2})$, if s_{i1} or s_{i2} has an extending story, then update the corresponding vector model with its related story to a new dynamic one. The generation procedure of dynamic vector will be described in next subsection.

   (b) Computing the cosine similarity between the two dynamic term vectors. If it exceeds the *extending threshold*, then s_{i1} and s_{i2} are the latest related stories for each other. If one story already has an extending story, replace the old one with the new one. So a story always has no more than one extending story at any time. If the similarity exceeds *topically link threshold*, s_{i1} and s_{i2} are topically linked.

From the above description, it is obvious that dynamic method needs two thresholds, one for making extending decision and the other for making link decision. Since in this paper we will focus on the optimum performance of systems, the first threshold is more important. But *topically link threshold* is also necessary to be properly defined to approach a better performance. In the baseline method, term vectors are dynamic because of the incremental $tf * idf$

weighting. However, dynamic information extending is another more important reason in the dynamic method. Whenever a story has an extending story, its vector representation will update to include the extending information. Having the extending method, the representation model can have more information to describe the topic in a story and make the topic evolve along with time. The dynamic method can define topic description clearer and get a more accurate similarity between stories.

5.3 Dynamic Vector Model

In the dynamic method, we have tried two ways for the generation of dynamic vector models: increment model and average model. Supposing we use vector model $M_1 = \{(f_{1i}, w_{1i}), i \geq 1\}$ of story s_1 to extend vector model $M_2 = \{(f_{2i}, w_{2i}), i \geq 1\}$ of story s_2, M_2 will change to representing the latest evolving topic described in current story after extending.

1. Increment Model: For each term f_{1i} in M_1, if it also occurs as f_{2j} in M_2, then w_{2j} will not change, otherwise (f_{1i}, w_{1i}) will be added into M_2. This dynamic vector model only takes interest in the new information that occurs only in M_1. For features both occurred in M_1 and M_2, the dynamic model will respect to their original weights.

2. Average Model: For each term f_{1i} in M_1, if it also occurs as f_{2j} in M_2, then $w_{2j} = 0.5 * (w_{1i} + w_{2j})$, otherwise (f_{1i}, w_{1i}) will be added into M_2. This dynamic model will take account of all information in M_1. So the difference between those two dynamic models is the weight recalculation method of the feature occurred in both M_1 and M_2.

Both the above two dynamic models can take account of information extending and topic evolution. Increment Model is closer to topic description since it is more dependent on latest term weights, while Average Model makes more reference to the centroid concept. The experiment results show that dynamic method with Average Model is a little superior to that with Increment Model.

6 Experiment and Discussion

6.1 Experiment Data

To evaluate the proposed method, we use the Chinese subset of TDT4 corpus (LDC, 2003) developed by the Linguistic Data Consortium (LDC) for TDT research. This subset contains 27145 stories all in Chinese from October 2000 through January 2001, which are gathered from news, broadcast or TV shows.

LDC totally labeled 40 topics on TDT4 for 2003 evaluation. There are totally 12334 stories pairs from 1151 source files in the experiment data. The answers for these pairs are based on 28 topics of these topics, generated from the LDC 2003 annotation documents. The first 2334 pairs are used for training and finding *extending threshold* of dynamic method. The rest 10000 pairs are testing data used for comparing performances of baseline and the dynamic methods.

6.2 Evaluation Measures

The work is measured by the TDT evaluation software, which could be referred to (Hoogma, 2005) for detail. Here is a brief description. The goal of link detection is to minimize the cost due to errors caused by the system. The TDT tasks are evaluated by computing a "detection cost":

$$C_{det} = C_{miss} \cdot P_{miss} \cdot P_{target} + C_{fa} \cdot P_{fa} \cdot P_{non-target}$$

where C_{miss} is the cost of a miss, P_{miss} is the estimated probability of a miss, P_{target} is the prior probability under which a pair of stories are linked, C_{fa} is the cost of a false alarm, P_{fa} is the estimated probability of a false alarm, and $P_{non-target}$ is the prior probability under which a pair of stories are not linked. A miss occurs when a linked story pair is not identified as being linked by the system. A false alarm occurs when the pair of stories that are not linked are identified as being linked by the system. A target is a pair of linked stories; conversely a non-target is a pair of stories that are not linked. For the link detection task these parameters are set as follows: C_{miss} is 1, P_{target} is 0.02, and C_{fa} is 0.1. The cost for each topic is equally weighted (usually the cost of topic-weighted is the mainly evaluation parameter) and normalized so that for a given system, the normalized value $(C_{det})_{norm}$ can be no less than

one without extracting information from the source data:

$$(C_{det})_{norm} = \frac{C_{det}}{\min(C_{miss}P_{target}, C_{fa}P_{non-target})}$$

$$(C_{det})_{overall} = \Sigma_i(C^i_{det})_{norm}/\#topics$$

where the sum is over topics i. A detection curve (DET curve) is computed by sweeping a threshold over the range of scores, and the minimum cost over the DET curve is identified as the minimum detection cost or min DET. The topic-weighted DET cost is dependent on both a good minimum cost and a good method for selecting an operating point, which is usually implemented by selecting a threshold. A system with a very low min DET cost can have a much larger topic-weighted DET score. Therefore, we focus on the minimum DET cost for the experiments.

6.3 Experiment Results

In this paper, we have tried three methods for story link detection: the baseline method described in Section 4 and two dynamic methods with different dynamic vectors introduced in Section 5. The following table gives their evaluation results.

metrics	baseline	dynamic 1	dynamic 2
P_{miss}	0.0514	0.0348	0.0345
P_{fa}	0.0067	0.0050	0.0050
$Clink_{min}$	0.0017	0.0012	0.0012
$Clink_{norm}$	0.0840	0.0591	0.0588

Table 1: Experiment Results of Baseline System and Dynamic Systems

In the table, $Clink_{min}$ is the minimum $(C_{det})_{overall}$, DET Graph Minimum Detection Cost (topic-weighted), $Clink_{norm}$ is the normalized minimum $(C_{det})_{overall}$, the dynamic 1 is the dynamic method which uses *Increment Model* and the dynamic 2 is the dynamic method which uses *Average Model*. We can see that the proposed two dynamic methods are both much better than baseline method on all four metrics. The $Clink_{Norm}$ of dynamic 1 and 2 are improved individually by 27.2% and 27.8% as compared to that of baseline method. The difference between two dynamic methods is due to different in the P_{miss}. However, it is too little to compare the two dynamic systems. We also make additional experiments in which a story is extended with all of its previous related stories. The minimum (Cdet)overall is 0.0614 for the system using Increment Model, and 0.0608 for the system using Average Model. Although the performances are also much superior to baseline, it is still a little poorer than that with only one latest related story, which confirm the ideal described in section 5.1.

Figure 1, 2 and 3 show the detail evaluation information for individual topic on Minimum Norm Detection Cost, P_{miss} and P_{fa}. From Figure 1 we know these two dynamic methods have improved the performance on almost all the topic, except topic 12, 26 and 32. Note that detection cost is a function of P_{miss} and P_{fa}. Figure 2 shows that both two dynamic methods reduce the false alarm rates on all evaluation topics. In Figure 3 there are 20 topics on which the miss rates remain zero or unchange. The dynamic methods reduce the miss rates on 5 topics. However, dynamic methods get relatively poorer results on topic 12, 26 and 32 . Altogether dynamic methods can notably improve system performance on evaluation metrics of both individual and weighted topic, especially the false alarm rate, but on some topics, it gets poorer results.

Further investigation shows that topic 12, 26 and 32 are about Presidential election in Ivory Coast on October 25, 2000, Airplane Crash in Chiang Kai Shek International Airport in Taiwan on October 31, 2000, and APEC Conference on November 12-15, 2000 at Brunei. After analyzing those story pairs with error link decision, we can split them into two sets. One is that two stories in a pair are general linked but not TDT specific topically linked. Here general linked means that there are many common words in two stories, but the events described in them happened in different times or different places. For example, Airplane Crash is a general topic, while Airplane Crash in certain location at specification time is a TDT topic. The other is that two stories in a pair are TDT topically linked while they describe the topic from different perspectives. In this condition they will have few common words. These may be due to that the information extracted from stories is still not accurate enough to represent them. It also may be because of the

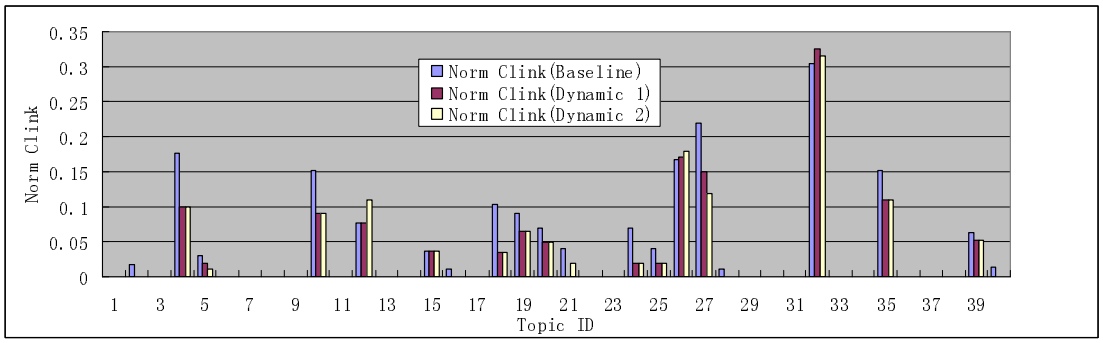


Figure 1: Normalized Minimum Detection Cost for individual topic

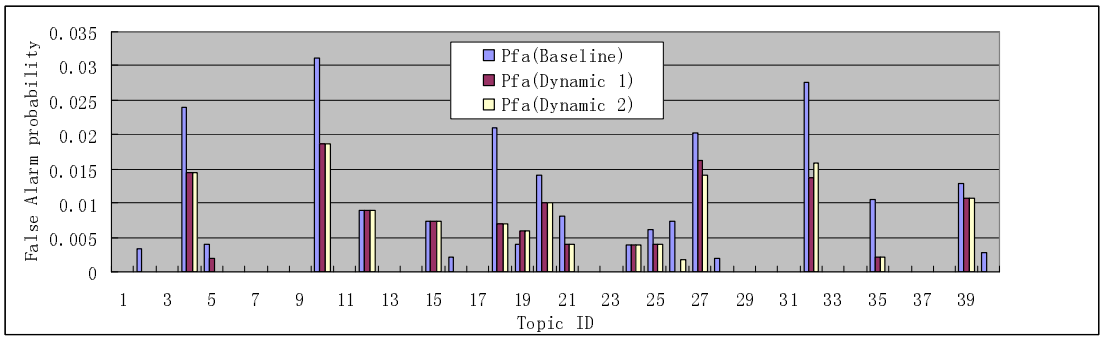


Figure 2: P_{fa} for individual topic

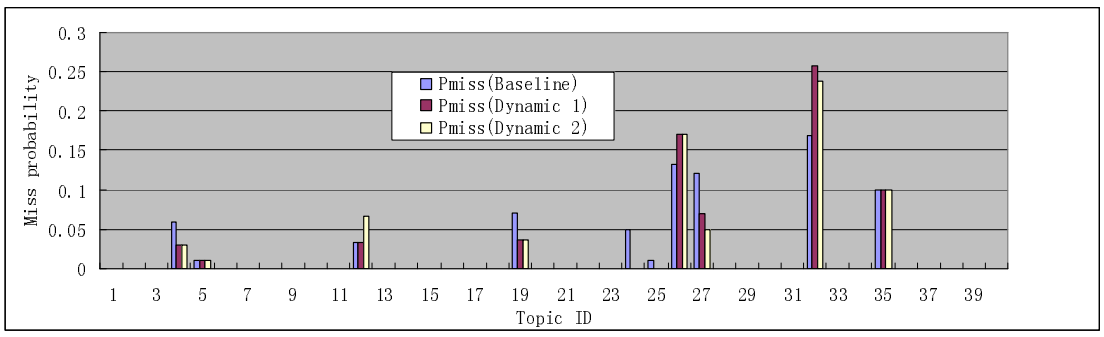


Figure 3: P_{miss} for individual topic

deficiency of vector model itself. Furthermore, we know that the extending story is chosen by cosine similarity, which results that the extending story and the extended story are usually topically linked from the same perspectives, seldom from different perspectives. Therefore the method of information extending may sometimes turn the above first problem worse and have no impact on the second problem. So mining more useful information or making more use of other useful resources to solve these problems will be the next work. In addition, how to represent this information with a proper model and seeking better or more proper representation models for TDT stories are also important issues. In a word, the method of information extending has been verified efficient in story link detection and can provide a reference to improve the performance of some other similar systems whose data must be processed serially, and it is also hopeful to combined with other improvement technologies.

7 Conclusion

Story link detection is a key technique in TDT research. Though many approaches have been tried, there are still some characters ignored. After analyzing the characters and deficiency in TDT stories and story link detection, this paper presents a method of dynamic information extending to improve the system performance by focus on two problems: information deficiency and topic evolution. The experiment results indicate that this method can effectively improve the performance on both miss and false alarm rates, especially the later one. However, we should realize that there are still some problems to solve in story link detection. How to compare general topically linked stories and how to compare stories describing a TDT topic from different angles will be very vital to improve system performance. The next work will focus on mining more and deeper useful information in TDT stories and exploiting more proper models to represent them.

Acknowledgement

This research is supported by the National Natural Science Foundation of China (60403050), Program for New Century Excellent Talents in University (NCET-06-0926) and the National Grand Fundamental Research Program of China under Grant(2005CB321802).

References

James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. 2000. Detections, bounds, and timelines: Umass and tdt–3. In *Proceedings of Topic Detection and Tracking (TDT–3)*, pages 167–174.

J. Allan, A. Bolivar, M. Connell, S. Cronen-Townsend, A Feng, F. Feng, G. Kumaran, L. Larkey, V. Lavrenko, and H. Raghavan. 2003. Umass tdt 2003 research summary. In *proceedings of TDT workshop*.

James Allan, editor. 2002. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, Norvell, Massachusetts.

Francine Chen, Ayman Farahat, and Thorsten Brants. 2004. Multiple similarity measures and source-pair information in story link detection. In *HLT-NAACL*, pages 313–320.

Margaret Connell, Ao Feng, Giridhar Kumaran, Hema Raghavan, Chirag Shah, and James Allan. 2004. Umass at tdt 2004. In *TDT2004 Workshop*.

Niek Hoogma. 2005. The modules and methods of topic detection and tracking. In *2nd Twente Student Conference on IT*.

Victor Lavrenko, James Allan, Edward DeGuzman, Daniel LaFlamme, Veera Pollard, and Stephen Thomas. 2002. Relevance models for topic detection and tracking. In *Proceedings of Human Language Technology Conference (HLT)*, pages 104–110.

LDC. 2003. Topic detection and tracking - phase 4. Technical report, Linguistic Data Consortium.

Ramesh Nallapati and James Allan. 2002. Capturing term dependencies using a language model based on sentence trees. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 383–390. ACM Press.

Ramesh Nallapati. 2003. Semantic language models for topic detection and tracking. In *HLT-NAACL*.

NIST. 2003. The 2003 topic detection and tracking task definition and evaluation plan. Technical report, National Institute of Standards and Technology(NIST).

Yiming Yang, Jian Zhang, Jaime Carbonell, and Chun Jin. 2002. Topic-conditioned novelty detection. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 688–693. ACM Press.

Orthographic Disambiguation Incorporating Transliterated Probability

Eiji ARAMAKI **Takeshi IMAI** **Kengo Miyo** **Kazuhiko Ohe**
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8655, Japan
aramaki@hcc.h.u-tokyo.ac.jp

Abstract

Orthographic variance is a fundamental problem for many natural language processing applications. The Japanese language, in particular, contains many orthographic variants for two main reasons: (1) transliterated words allow many possible spelling variations, and (2) many characters in Japanese nouns can be omitted or substituted. Previous studies have mainly focused on the former problem; in contrast, this study has addressed both problems using the same framework. First, we automatically collected both positive examples (sets of equivalent term pairs) and negative examples (sets of inequivalent term pairs). Then, by using both sets of examples, a support vector machine based classifier determined whether two terms (t_1 and t_2) were equivalent. To boost accuracy, we added a transliterated probability $P(t_1|s)P(t_2|s)$, which is the probability that both terms (t_1 and t_2) were transliterated from the same source term (s), to the machine learning features. Experimental results yielded high levels of accuracy, demonstrating the feasibility of the proposed approach.

1 Introduction

Spelling variations, such as "*center*" and "*centre*", which have different spellings but identical meanings, are problematic for many NLP applications including information extraction (IE), question answering (QA), and machine transliteration (MT). In this paper, these variations can be termed **orthographic variants**.

Table 1: Examples of Orthographic Variants.

spaghetti	Thompson operation
スパゲッティ ⟨ supagettji ⟩ スパゲッティー ⟨ supagettjii ⟩ スパゲティ ⟨ supagetji ⟩ スパゲティー ⟨ supagetjii ⟩ スパゲテイ ⟨ supagetji ⟩	トンプソンの手術法 (Thompson's operation method) トンプソンの手術 (Thompson's operation) トンプソン術 (Thompson operation)

* ⟨⟩ indicates a pronunciation. () indicates a translation.

The Japanese language, in particular, contains many orthographic variants, for two main reasons:

1. It imports many words from other languages using transliteration, resulting in many possible spelling variations. For example, Masuyama et al. (2004) found at least six different spellings for " spaghetti " in newspaper articles (Table 1 Left).

2. Many characters in Japanese nouns can be omitted or substituted, leading to tons of insertion variations (Daille et al., 1996) (Table 1 Right).

To address these problems, this study developed a support vector machine (SVM) based classifier that

can determine whether two terms are equivalent. Because a SVM-based approach requires positive and negative examples, we also developed a method to automatically generate both examples.

Our proposed method differs from previously developed methods in two ways.

1. Previous studies have focused solely on the former problem (transliteration); our target scope is wider. We addressed both transliteration and character omissions/substitutions using the same framework.

2. Most previous studies have focused on back-transliteration (Knight and Graehl, 1998; Goto et al., 2004), which has the goal of generating a source word (s) for a Japanese term (t). In contrast, we employed a discriminative approach, which has the goal of determining whether two terms (t_1 and t_2) are equivalent. These two goals are related. For example, if two terms (t_1 and t_2) were transliterated from the same word (s), they should be orthographic variants. To incorporate this information, we incorporated a transliterated-probability ($P(s|t_1) \times P(s|t_2)$) into the SVM features.

Although we investigated performance using medical terms, our proposed method does not depend on a target domain[1].

2 Orthographic Variance in Dictionary Entries

Before developing our methodology, we examined problems related to orthographic variance.

First, we investigated the amount of orthographic variance between two dictionaries' entries (DIC1 (Ito et al., 2003), totaling 69,604 entries, and DIC2 (Nanzando, 2001), totaling 27,971 entries).

Exact matches between entries only occurred for 10,577 terms (15.1% of DIC1, and 37.8% of DIC2). From other entries, we extracted orthographic variance as follows.

STEP 1: Extracting Term Pairs with Similar Spelling

We extracted term pairs with similar spelling (t_1 and t_2) using edit distance-based similarity (defined by Table 2). We extracted term pairs with $SIM_{ed} > 0.8$, and found 5,064 term pairs with similar spelling.

STEP 2: Judging Orthographic Variance

We then manually judged whether each term pair was composed of orthographic variants (whether or not they had the same meaning).

Our results indicated that 1,889 (37.3%) of the terms were orthographic variants.

Figure 1 presents the relation between the orthographic variation ratio and similarity threshold (0.8-1.0). As shown in the figure, a higher similarity threshold (SIM=0.96-97) does not always indicate that terms are orthographic variants.

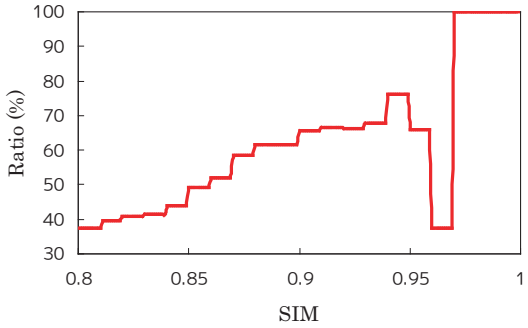


Figure 1: Similarity Threshold and Orthographic Variants Ratio.

The following term pair is a typical example:

1. 変異 **B** 型肝炎ウィルス
   (*mutated hepatitis type* **B** *virus*),

2. 変異 **C** 型肝炎ウィルス
   (*mutated hepatitis type* **C** *virus*).

They have only one character difference ("**B**" and "**C**"), resulting in high levels of spelling similarity, but the meanings are not equivalent. This type of limitation, intrinsic to measurements of spelling similarity, motivated us to develop an SVM-based classifier.

3 Method

We developed an SVM-based classifier that determines whether two terms are equivalent. Section 3.1

[1]The domain could affect the performance, because most of medical terms are imported from other languages, leading to many orthographic variants.

Table 2: Edit Distance-based Similarity (SIM_{ed}).

The edit distance-based similarity (SIM_{ed}) between two terms $(t_1, t2)$ is defined as follows:

$$SIM_{ed}(t_1, t_2) = 1 - \frac{\text{EditDistance}(t_1, t_2) \times 2}{len(t_1) + len(t_2)},$$

where $len(t_1)$ is the number of characters of t_1, $len(t_2)$ is the number of characters of t_2, Edit Distance(t_1, t_2) is the minimum number of point mutations required to change t_1 into t_2, where a point mutation is one of: (1) a change in a character, (2) the insertion of a character, and (3) the deletion of a character. For details, see (Levenshtein, 1965).

will describe the method we used to build training data, and Section 3.2 will introduce the classifier.

3.1 Automatic Building of Examples

Positive Examples

Our method uses a straight forward approach to extract positive examples. The basic idea is that orthographic variants should have (1) similar spelling, and (2) the same English translation.

The method consists of the following two steps:

STEP 1: First, using two or more translation dictionaries, extract a set of Japanese terms with the same English translation.

STEP 2: Then, for each extracted set, generate two possible term pairs (t_1 and t_2) and calculate the spelling similarity between them. Spelling similarity is measured by edit distance-based similarity (see Section 2). Any term pair with more than a threshold ($SIMed(t_1, t_2) > 0.8$) similarity is considered a positive example.

Negative Examples

We based our method of extracting negative examples using the dictionary-based method. As with positive examples, we collected term pairs with similar spellings ($SIMed(t_1, t_2) > 0.8$), but differing English translations.

However, the above heuristic is not sufficient to extract negative examples; different English terms might have the same meaning, which could cause unsuitable negative examples.

For example, t_1 "胃癌 (*stomach cancer*)" and t_2 "胃がん (*stomach carcinoma*)": although these words have differing English translations, unfortunately they are not a negative example ("*cancer*" and "*carcinoma*" are synonymous).

To address this problem, we employed a corpus-based approach, hypothesizing that if two terms are orthographic variants, they should rarely both appear in the same document. Conversely, if both terms appear together in many documents, they are unlikely to be orthographic variants (negative examples).

Based on this assumption, we defined the following scoring method:

$$Score(t_1, t_2) = \frac{\log(HIT(t_1, t_2))}{max(\log(HIT(t_1)), \log(HIT(t_2)))},$$

where $HIT(t)$ is the number of Google hits for a query t. We only used negative examples with the highest K score, and discarded the others[2].

3.2 SVM-Based Classifier

The next problem was how to convert training-data into machine learning features. We used two types of features.

Character-Based Features

We expressed different characters between two terms and their context (window size ± 1) as features, shown in Table 3. Thus, to represent an omission, "ϕ (*null*)" is considered a character. Two examples are provided in Figures 2.

Note that if terms contain two or more differing parts, all the differing parts are converted into features.

Similarity-based Features

Another type of feature is the similarity between two terms (t_1 and t_2). We employed two similarities:

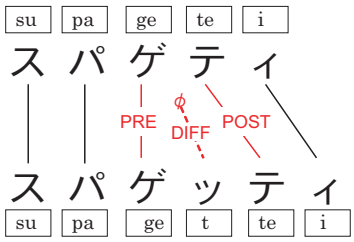
1. Edit distance-based similarity $SIM_{ed}(t_1, t_2)$ (see Section 2).

2. Transliterated similarity, which is the probability that two terms (t_1 and t_2) were transliterated

[2] In the experiments in Section 4, we set K is 41,120, which is equal to the number of positive examples.

Table 3: Character-based Features.

LEX-DIFF	Differing characters between two terms, consisting of a pair of $n : m$ characters ($n > 0$ and $m > 0$). For example, we regard "ッ(t)→ ϕ" as LEX-DIFF in Figure 2 TOP.
LEX-PRE	Previous character of DIFF. We regard "ゲ(ge)" as LEX-PRE in Figure 2 TOP.
LEX-POST	Subsequent character of DIFF. We regard "テ(te)" as LEX-POST in Figure 2 TOP.
TYPE-DIFF	A script type of differing characters between two terms, classified into four categories: (1) HIRAGANA-script, (2) KATAKANA-script, (3) Chinese-character script or (4) others (symbols, numerous expressions etc.)) We regard "KATAKANA→ ϕ" as TYPE-DIFF in Figure 2 TOP.
TYPE-PRE	A type previous character of DIFF. We regard "KATAKANA" as TYPE-PRE in Figure 2 TOP.
TYPE-POST	A type subsequent character of DIFF. We regard "KATAKANA" as TYPE-POST in Figure 2 TOP.
LEN-DIFF	A length (the number of characters) of differing parts.



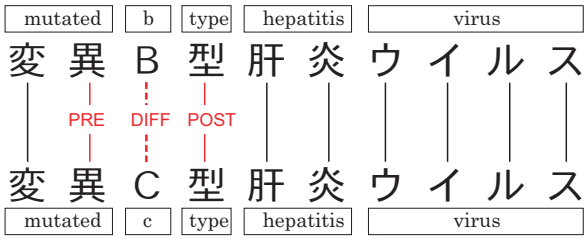


Figure 2: A Positive Example (TOP) and A Negative Example (BOTTOM).

from the same source word (t) (defined in Table 4).

Note that the latter, transliterated similarity, is applicable to a situation in which the input pair is transliterated.

4 Experiments

4.1 Test-Set

To evaluate the performance of our system, we used judged term pairs, as discussed in Section 2 (ALL-SET). We also extracted a sub-set of these pairs in order to focus on a transliteration problem (TRANS-SET).

1. **ALL-SET**: This set consisted of all examples (1,889 orthographic variants of 5,064 pairs)

2. **TRANS-SET**: This set contained only examples of transliteration (543 orthographic variants or 1,111 pairs).

4.2 Training-Set

Using the proposed method set out in Section 3, we automatically constructed a training-set from two translation dictionaries (Japan Medical Terminology English-Japanese(Nanzando, 2001) and 25-Thousand-Term Medical Dictionary(MEID, 2005)).

The resulting training-set consisted of 82,240 examples (41,120 positive examples and 41,120 negative examples).

4.3 Comparative Methods

We compared the following methods:

1. **SIM-ED**: An edit distance-based method, which regards an input with a similarity $SIM_{ed}(t_1, t_2) > TH$ as an orthographic variant.

2. **SIM-TR**: A transliterated based method, which regards an input with a spelling similarity $SIM_{tr}(t_1, t_2) > TH$ as an orthographic variant (TRANS-SET only).

3. **PROPOSED**: Our proposed method without SIM_{tr} features.

4. **PROPOSED+TR**: Our proposed method with SIM_{tr} features. (TRANS-SET only).

For SVM learning, we used TinySVM[3] with polynomial kernel (d=2).

4.4 Evaluation

We used the three following measures to evaluate our method:

$$Precision = \frac{\text{\# of pairs found and correct}}{\text{total \# of pairs found}},$$

$$Recall = \frac{\text{\# of pairs found and correct}}{\text{total \# of pairs correct}},$$

$$F_{\beta=1} = 2 \times \frac{Recall \times Precision}{Recall + Precision}.$$

4.5 Results

Table 5 presents the performance of all methods. The accuracy of similarity-based methods (SIM-ED and SIM-TR) varied depending on the threshold (TH). Figure 3 is a precision-recall graph of all methods in TRANS-SET.

In ALL-SET, PROPOSED outperformed a similarity-based method (SIM-ED) in $F_{\beta=1}$, demonstrating the feasibility of the proposed discriminative approach.

[3]http://chasen.org/ taku/software/TinySVM/

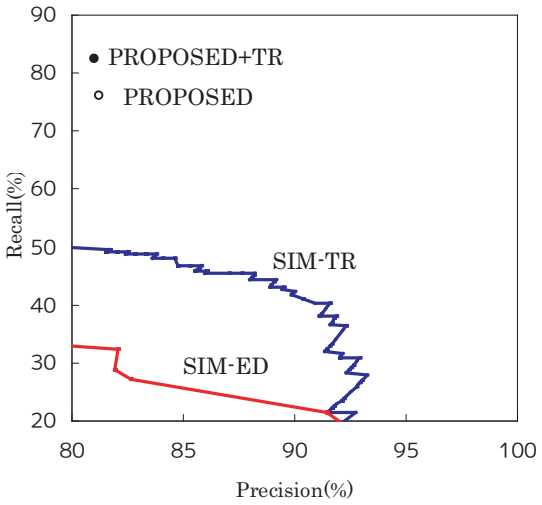


Figure 3: SIM and orthographic variants ratio.

In TRANS-SET, PROPOSED also outperformed two similarity-based methods (SIM-ED and SIM-TR). In addition, PROPOSED+TR yielded higher levels of accuracy than PROPOSED. Based on this result, we can conclude that adding transliterated-probability improved accuracy.

It was difficult to compare accuracy between the results of our study and previous studies. Previous studies used different corpora, and also focused on (back-) transliteration. However, our accuracy levels were at least as good as those in previous studies (64% by (Knight and Graehl, 1998) and 87.7% by (Goto et al., 2004)).

4.6 Error Analysis

We investigated errors from PROPOSED and PROPOSED+TR, and found two main types.

1. **Different Script Types**
   The Japanese language can be expressed using three types of script: KANJI (Chinese characters), KATAKANA, and HIRAGANA. Although each of these scripts can be converted to another, (such as "癲癇" ("*epilepsia*" in KANJI script) and "てんかん" ("*epilepsia*" in HIRAGANA script), our method cannot deal with this phenomenon. Future research will need to add steps to solve this problem.

2. **Transliteration from Non-English Lan-**

Table 5: Results

	ALL-SET			TRANS-SET		
	Precision	Recall	$F_{\beta=1}$	Precision	Recall	$F_{\beta=1}$
SIM-ED	65.2%	64.6%	0.65	91.2%	36.3%	0.51
SIM-TR	-	-	-	**92.6%**	43.9%	0.59
PROPOSED	78.2%	70.2%	0.73	81.9%	75.6%	0.78
PROPOSED+TR	-	-	-	81.7%	**82.7%**	**0.82**

* The performance in SIM-ED and SIM-TR showed the highest $F_{\beta=1}$ values.

guages

While our experimental set consisted of medical terms, including a few transliterations from Latin or German, transliteration-probability was trained using transliterations from the English language (using a general dictionary). Therefore, PROPOSED+TR results are inferior when inputs are from non-English languages. In a general domain, SIM-TR and PROPOSED+TR would probably yield higher accuracy.

5 Related Works

As noted in Section 1, transliteration is the most relevant field to our work, because it results in many orthographic variations.

Most previous transliteration studies have focused on finding the most suitable back-transliteration of a term. For example, Knight (1998) proposed a probabilistic model for transliteration. Goto et al.(2004) proposed a similar method, utilizing surrounding characters.

Their method is not only applicable to Japanese; it has already been used for Korean(Oh and Choi, 2002; Oh and Choi, 2005; Oh and Isahara, 2007), Arabic(Stalls and Knight, 1998; Sherif and Kondrak, 2007), Chinese(Li et al., 2007), and Persian(Karimi et al., 2007).

Our method uses a different kind of task-setting, compared to previous methods. It is based on determining whether two terms within the same language are equivalent. It provides high levels of accuracy, which should be practical for many applications.

Another issue is that of how to represent transliteration phenomena. Methods can be classified into three main types: grapheme-based (Li et al., 2004); phoneme-based (Knight and Graehl, 1998); and combinations of both these methods(hybrid-model(Bilac and Tanaka, 2004) and correspondence-based model(Oh and Choi, 2002; Oh and Choi, 2005)). Our proposed method employed a grapheme-based approach. We selected this kind of approach because it allows us to handle not only transliteration but also character omissions/substitutions, which we would not be able to address using a phoneme-based approach (and a combination approach).

Yoon et al. (2007) also proposed a discriminative transliteration method, but their system was based on determining whether a target term was transliterated from a source term.

Bergsma and Kondrak (2007) and Aramaki et al. (2007) proposed on a discriminative method for similar spelling terms. However, they did not deal with a transliterated probability.

Masuyama et al. (2004) collected 178,569 Japanese transliteration variants (positive examples) from a large corpus. In contrast, we collected both positive and negative examples in order to train the classifier.

6 Conclusion

We developed an SVM-based orthographic disambiguation classifier, incorporating transliteration probability. We also developed a method for collecting both positive and negative examples. Experimental results yielded high levels of accuracy, demonstrating the feasibility of the proposed approach. Our proposed classifier could become a fundamental technology for many NLP applications.

Acknowledgments

Part of this research is supported by Grant-in-Aid for Scientific Research of Japan So-

Table 4: Transliterated Similarity (SIM_{tr}).

The transliterated similarity (SIM_{tr}) between two terms $(t_1, t2)$ is defined as follows[a]:

$$SIM_{tr}(t_1, t_2) = \sum_{s \in S} P(t_1|s)P(t_2|s),$$

where S is a set of back-transliterations that are generated from both t_1 and t_2, $P(e|t)$ is a probability of Japanese term (t) comes from a source term s.

$$P(t|s) = \prod_{k=1}^{|K|} P(t_k|s_k),$$

$$P(t_k|s_k) = \frac{\text{frequency of } s_k \rightarrow t_k}{\text{frequency of } s_k},$$

where $|K|$ is the number of characters in a term t, t_k is the k-th character of a term t, s_k is the k-th character sequence of a term s, "frequency of $s_k \rightarrow t_k$" is the occurrences of the alignments, "frequency of s_k" is the occurrences of a character s_k.

To get alignment, we extracted 100,128 transliterated term pairs from a transliteration dictionary (EDP, 2005), and estimate its alignment by using GIZA++[b]. We aligned in Japanese-to-English direction, and got $1 : m$ alignments (one Japanese character : m alphabetical characters) to calculate $P(t_k|s_k)$. These formulas are equal to (Karimi et al., 2007).

[a] $SIM_{tr}(t_1, t_2)$ is a similarity (not a probability)
[b] http://www.fjoch.com/GIZA++.html

ciety for the Promotion of Science (Project Number:16200039, F.Y.2004-2007 and 18700133, F.Y.2006-2007) and the Research Collaboration Project (#047100001247) with Japan Anatomy Laboratory Co.Ltd.

References

Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. 2007. Support vector machine based orthographic disambiguation. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation (TMI2007)*, pages 21–30.

Shane Bergsma and Grzegorz Kondrak. 2007. Alignment-based discriminative string similarity. In *Proceedings of the Association for Computational Linguistics (ACL2007)*, pages 656–663.

Slaven Bilac and Hozumi Tanaka. 2004. A hybrid back-transliteration system for Japanese. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 597–603.

B. Daille, B. Habert, C. Jacquemin, and J. Royaut. 1996. Empirical observation of term variations and principles for their description. *Terminology*, 3(2):197–258.

EDP. 2005. Eijiro Japanese-English dictionary, electronic dictionary project.

Isao Goto, Naoto Kato, Terumasa Ehara, and Hideki Tanaka. 2004. Back transliteration from Japanese to English using target English context. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 827–833.

M. Ito, H. Imura, and H. Takahisa. 2003. *IGAKUSHOIN'S MEDICAL DICTIONARY*. Igakusyoin.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2007. Collapsed consonant and vowel models: New approaches for English-Persian transliteration and back-transliteration. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 648–655.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

V. I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the Meeting of the Association for Computational Linguistics (ACL2004)*, pages 159–166.

Haizhou Li, Khe Chai Sim, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic transliteration of personal names. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 120–127.

Takeshi Masuyama, Satoshi Sekine, and Hiroshi Nakagawa. 2004. Automatic construction of Japanese KATAKANA variant list from large corpus. In *Proceedings of The 20th International Conference on Computational Linguistics (COLING2004)*, pages 1214–1219.

MEID. 2005. *25-Mango Medical Dictionary*. Nichigai Associates, Inc.

Nanzando. 2001. *Japan Medical Terminology English-Japanese 2nd Edition*. Committee of Medical Terminology, NANZANDO Co.,Ltd.

Jong-Hoon Oh and Key-Sun Choi. 2002. An English-Korean transliteration model using pronunciation and contextual rules. In *Proceedings of The 19th International Conference on Computational Linguistics (COLING2002)*, pages 758–764.

Jong-Hoon Oh and Key-Sun Choi. 2005. An ensemble of grapheme and phoneme for machine transliteration. In *Proceedings of Second International Joint Conference on Natural Language Processing (IJCNLP2005)*, pages 450–461.

Jong-Hoon Oh and Hitoshi Isahara. 2007. Machine transliteration using multiple transliteration engines and hypothesis re-ranking. In *Proceedings of MT Summit XI*, pages 353–360.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 944–951.

Bonnie Glover Stalls and Kevin Knight. 1998. Translating names and technical terms in arabic text. In *Proceedings of The International Conference on Computational Linguistics and the 36th Annual Meeting of the Association of Computational Linguistics (COLING-ACL1998) Workshop on Computational Approaches to Semitic Languages*.

Su-Youn Yoon, Kyoung-Young Kim, and Richard Sproat. 2007. Multilingual transliteration using feature based phonetic method. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL2007)*, pages 112–119.

Name Origin Recognition Using Maximum Entropy Model and Diverse Features

Min Zhang[1], Chengjie Sun[2], Haizhou Li[1], Aiti Aw[1], Chew Lim Tan[3], Xiaolong Wang[2]

[1]Institute for Infocomm Research, Singapore
{mzhang,hli,aaiti}@i2r.a-star.edu.sg

[2]Harbin Institute of Technology, China
{cjsun,wangxl}@insun.hit.edu.cn

[3]National University of Singapore, Singapore
tancl@comp.nus.edu.sg

Abstract

Name origin recognition is to identify the source language of a personal or location name. Some early work used either rule-based or statistical methods with single knowledge source. In this paper, we cast the name origin recognition as a multi-class classification problem and approach the problem using Maximum Entropy method. In doing so, we investigate the use of different features, including phonetic rules, *n*-gram statistics and character position information for name origin recognition. Experiments on a publicly available personal name database show that the proposed approach achieves an overall accuracy of 98.44% for names written in English and 98.10% for names written in Chinese, which are significantly and consistently better than those in reported work.

1 Introduction

Many technical terms and proper names, such as personal, location and organization names, are translated from one language into another with approximate phonetic equivalents. The phonetic translation practice is referred to as *transliteration*; conversely, the process of recovering a word in its native language from a transliteration is called as *back-transliteration* (Zhang et al, 2004; Knight and Graehl, 1998). For example, English name "Smith" and "史密斯 (*Pinyin*[1]: Shi-Mi-Si)" in Chinese form a pair of transliteration and *back-transliteration*. In many natural language processing tasks, such as machine translation and cross-lingual information retrieval, automatic name transliteration has become an indispensable component.

Name origin refers to the source language of a name where it originates from. For example, the origin of the English name "Smith" and its Chinese transliteration "史密斯 (Shi-Mi-Si)" is English, while both "Tokyo" and "东京 (Dong-Jing)" are of Japanese origin. Following are examples of different origins of a collection of English-Chinese transliterations.

English:	Richard-理查德 (Li-Cha-De)
	Hackensack-哈肯萨克(Ha-Ken-Sa-Ke)
Chinese:	Wen JiaBao-温家宝(Wen-Jia-Bao)
	ShenZhen–深圳(Shen-Zhen)
Japanese:	Matsumoto-松本 (Song-Ben)
	Hokkaido-北海道(Bei-Hai-Dao)
Korean:	Roh MooHyun-卢武铉(Lu-Wu-Xuan)
	Taejon-大田(Da-Tian)
Vietnamese:	Phan Van Khai-潘文凯(Pan-Wen-Kai)
	Hanoi-河内(He-Nei)

In the case of machine transliteration, the name origins dictate the way we re-write a foreign word. For example, given a name written in English or Chinese for which we do not have a translation in

[1] *Hanyu Pinyin*, or *Pinyin* in short, is the standard romanization system of Chinese. In this paper, *Pinyin* is given next to Chinese characters in round brackets for ease of reading.

a English-Chinese dictionary, we first have to decide whether the name is of Chinese, Japanese, Korean or some European/English origins. Then we follow the transliteration rules implied by the origin of the source name. Although all English personal names are rendered in 26 letters, they may come from different romanization systems. Each romanization system has its own rewriting rules. English name "Smith" could be directly transliterated into Chinese as "史密斯(Shi-Mi-Si)" since it follows the English phonetic rules, while the Chinese translation of Japanese name "Koizumi" becomes "小泉(Xiao-Quan)" following the Japanese phonetic rules. The name origins are equally important in back-transliteration practice. Li et al. (2007) incorporated name origin recognition to improve the performance of personal name transliteration. Besides multilingual processing, the name origin also provides useful semantic information (regional and language information) for common NLP tasks, such as co-reference resolution and name entity recognition.

Unfortunately, little attention has been given to name origin recognition (NOR) so far in the literature. In this paper, we are interested in two kinds of name origin recognition: the origin of names written in English (**ENOR**) and the origin of names written in Chinese (**CNOR**). For ENOR, the origins include English (Eng), Japanese (Jap), Chinese Mandarin *Pinyin* (Man) and Chinese Cantonese *Jyutping* (Can). For CNOR, they include three origins: Chinese (Chi, for both Mandarin and Cantonese), Japanese and English (refer to Latin-scripted language).

Unlike previous work (Qu and Grefenstette, 2004; Li et al., 2006; Li et al., 2007) where NOR was formulated with a generative model, we regard the NOR task as a classification problem. We further propose using a discriminative learning algorithm (Maximum Entropy model: MaxEnt) to solve the problem. To draw direct comparison, we conduct experiments on the same personal name corpora as that in the previous work by Li *et al.* (2006). We show that the MaxEnt method effectively incorporates diverse features and outperforms previous methods consistently across all test cases.

The rest of the paper is organized as follows: in section 2, we review the previous work. Section 3 elaborates our proposed approach and the features. Section 4 presents our experimental setup and reports our experimental results. Finally, we conclude the work in section 5.

2 Related Work

Most of previous work focuses mainly on ENOR although same methods can be extended to CNOR. We notice that there are two informative clues that used in previous work in ENOR. One is the lexical structure of a romanization system, for example, Hanyu *Pinyin*, Mandarin Wade-Giles, Japanese Hepbrun or Korean Yale, each has a finite set of syllable inventory (Li et al., 2006). Another is the phonetic and phonotactic structure of a language, such as phonetic composition, syllable structure. For example, English has unique consonant clusters such as /str/ and /ks/ which Chinese, Japanese and Korean (CJK) do not have. Considering the NOR solutions by the use of these two clues, we can roughly group them into two categories: rule-based methods (for solutions based on lexical structures) and statistical methods (for solutions based on phonotactic structures).

Rule-based Method

Kuo and Yang (2004) proposed using a rule-based method to recognize different romanization system for Chinese only. The left-to-right longest match-based lexical segmentation was used to parse a test word. The romanization system is confirmed if it gives rise to a successful parse of the test word. This kind of approach (Qu and Grefenstette, 2004) is suitable for romanization systems that have a finite set of *discriminative* syllable inventory, such as *Pinyin* for Chinese Mandarin. For the general tasks of identifying the language origin and romanization system, rule based approach sounds less attractive because not all languages have a finite set of *discriminative* syllable inventory.

Statistical Method

1) N-gram Sum Method (SUM): Qu and Grefenstette (2004) proposed a NOR identifier using a trigram language model (Cavnar and Trenkle, 1994) to distinguish personal names of three language origins, namely Chinese, Japanese and English. In their work, the training set includes 11,416 Chinese name entries, 83,295 Japanese name entries and 88,000 English name entries. However, the trigram is defined as the joint probabil-

ity $p(c_i c_{i-1} c_{i-2})$ for 3-character $c_i c_{i-1} c_{i-2}$ rather than the commonly used conditional probability $p(c_i \mid c_{i-1} c_{i-2})$. Therefore, the so-called trigram in Qu and Grefenstette (2004) is basically a substring unigram probability, which we refer to as the n-gram (n-character) sum model (SUM) in this paper. Suppose that we have the unigram count $C(c_i c_{i-1} c_{i-2})$ for character substring $c_i c_{i-1} c_{i-2}$, the unigram is then computed as:

$$p(c_i c_{i-1} c_{i-2}) = \frac{C(c_i c_{i-1} c_{i-2})}{\sum_{i, c_i c_{i-1} c_{i-2}} C(c_i c_{i-1} c_{i-2})} \quad (1)$$

which is the count of character substring $c_i c_{i-1} c_{i-2}$ normalized by the sum of all 3-character string counts in the name list for the language of interest. For origin recognition of Japanese names, this method works well with an accuracy of 92%. However, for English and Chinese, the results are far behind with a reported accuracy of 87% and 70% respectively.

2) N-gram Perplexity Method (PP): Li *et al.* (2006) proposed using n-gram character perplexity PP_c to identify the origin of a Latin-scripted name. Using bigram, the PP_c is defined as:

$$PP_c = 2^{-\frac{1}{N_c} \sum_{i=1}^{N_c} \log p(c_i \mid c_{i-1})} \quad (2)$$

where N_c is the total number of characters in the test name, c_i is the i^{th} character in the test name. $p(c_i \mid c_{i-1})$ is the bigram probability which is learned from each name list respectively. As a function of model, PP_c measures how good the model matches the test data. Therefore, PP_c can be used to measure how good a test name matches a training set. A test name is identified to belong to a language if the language model gives rise to the minimum perplexity. Li *et al.* (2006) shown that the PP method gives much better performance than the SUM method. This may be due to the fact that the PP measures the normalized conditional probability rather than the sum of joint probability. Thus, the PP method has a clearer mathematical interpretation than the SUM method.

The statistical methods attempt to overcome the shortcoming of rule-based method, but they suffer from data sparseness, especially when dealing with a large character set, such as in Chinese (our experiments will demonstrate this point empirically). In this paper, we propose using Maximum Entropy (MaxEnt) model as a general framework for both ENOR and CNOR. We explore and integrate multiple features into the discriminative classifier and use a common dataset for benchmarking. Experimental results show that the MaxEnt model effectively incorporates diverse features to demonstrate competitive performance.

3 MaxEnt Model and Features

3.1 MaxEnt Model for NOR

The principle of maximum entropy (MaxEnt) model is that given a collection of facts, choose a model consistent with all the facts, but otherwise as uniform as possible (Berger *et al.*, 1996). MaxEnt model is known to easily combine diverse features. For this reason, it has been widely adopted in many natural language processing tasks. The MaxEnt model is defined as:

$$p(c_i \mid x) = \frac{1}{Z} \prod_{j=1}^{K} \alpha_j^{f_j(c_i, x)} \quad (3)$$

$$Z = \sum_{i=1}^{N} p(c_i \mid x) = \sum_{i=1}^{N} \prod_{j=1}^{K} \alpha_j^{f_j(c_i, x)} \quad (4)$$

where c_i is the outcome label, x is the given observation, also referred to as an instance. Z is a normalization factor. N is the number of outcome labels, the number of language origins in our case. $f_1, f_2, \cdots, f_K$ are feature functions and $\alpha_1, \alpha_2, \cdots, \alpha_K$ are the model parameters. Each parameter corresponds to exactly one feature and can be viewed as a "weight" for the corresponding feature.

In the NOR task, c is the name origin label; x is a personal name, f_i is a feature function. All features used in the MaxEnt model in this paper are binary. For example:

$$f_j(c, x) = \begin{cases} 1, & \text{if } c = \text{"Eng"} \,\&\, x \text{ contains("str")} \\ 0, & \text{otherwise} \end{cases}$$

In our implementation, we used Zhang's maximum entropy package[2].

3.2 Features

Let us use English name "Smith" to illustrate the features that we define. All characters in a name

[2] http://homepages.inf.ed.ac.uk/s0450736/maxent.html

are first converted into upper case for ENOR before feature extraction.

N-gram Features: N-gram features are designed to capture both phonetic and orthographic structure information for ENOR and orthographic information only for CNOR. This is motivated by the facts that: 1) names written in English but from non-English origins follow different phonetic rules from the English one; they also manifest different character usage in orthographic form; 2) names written in Chinese follows the same pronunciation rules (*Pinyin*), but the usage of Chinese characters is distinguishable between different language origins as reported in Table 2 of (Li *et al.*, 2007). The N-gram related features include:

1) FUni: character unigram <*S, M, I, T, H*>
2) FBi: character bigram <*SM, MI, IT, TH*>
3) FTri: character trigram <*SMI, MIT, ITH* >

Position Specific n-gram Features: We include position information into the n-gram features. This is mainly to differentiate surname from given name in recognizing the origin of CJK personal names written in Chinese. For example, the position specific n-gram features of a Chinese name "温家宝(Wen-Jia-Bao)" are as follows:

1) FPUni: position specific unigram <*0 温*(Wen), *1 家*(Jia), *2 宝*(Bao)>
2) FPBi: position specific bigram <*0 温家*(Wen-Jia), *1 家宝*(Jia-Bao)>
3) FPTri: position specific trigram <*0 温家宝*(Wen-Jia-Bao)>

Phonetic Rule-based Features: These features are inspired by the rule-based methods (Kuo and Yang, 2004; Qu and Grefenstette, 2004) that check whether an English name is a sequence of syllables of CJK languages in ENOR task. We use the following two features in ENOR task as well.

1) FMan: a Boolean feature to indicate whether a name is a sequence of Chinese Mandarin *Pinyin*.
2) FCan: a Boolean feature to indicate whether a name is a sequence of Cantonese *Jyutping*.

Other Features:

1) FLen: the number of Chinese characters in a given name. This feature is for CNOR only. The numbers of Chinese characters in personal names vary with their origins. For example, Chinese and Korean names usually consist of 2 to 3 Chinese characters while Japanese names can have up to 4 or 5 Chinese characters
2) FFre: the frequency of n-gram in a given name. This feature is for ENOR only. In CJK names, some consonants or vowels usually repeat in a name as the result of the regular syllable structure. For example, in the Chinese name "Zhang Wanxiang", the bigram "an" appears three times

Please note that the trigram and position specific trigram features are not used in CNOR due to anticipated data sparseness in CNOR[3].

4 Experiments

We conduct the experiments to validate the effectiveness of the proposed method for both ENOR and CNOR tasks.

4.1 Experimental Setting

Origin	# entries	Romanization System
Eng[4]	88,799	English
Man[5]	115,879	*Pinyin*
Can	115,739	*Jyutping*
Jap[6]	123,239	Hepburn

Table 1: D_E: Latin-scripted personal name corpus for ENOR

Origin	# entries
Eng[7]	37,644
Chi[8]	29,795
Jap[9]	33,897

Table 2: D_C: Personal name corpus written in Chinese characters for CNOR

[3] In the test set of CNOR, 1080 out of 2980 names of Chinese origin do not consist of any bigrams learnt from training data, while 2888 out of 2980 names do not consist of any learnt trigrams. This is not surprising as most of Chinese names only have two or three Chinese characters and in our open testing, the train set is exclusive of all entries in the test set.

[4] http://www.census.gov/genealogy/names/

[5] http://technology.chtsai.org/namelist/

[6] http://www.csse.monash.edu.au/~jwb/enamdict_doc.html

[7] Xinhua News Agency (1992)

[8] http://www.ldc.upenn.edu LDC2005T34

[9] www.cjk.org

Datasets: We prepare two data sets which are collected from publicly accessible sources: D_E and D_C for the ENOR and CNOR experiment respectively. D_E is the one used in (Li *et al.*, 2006), consisting of personal names of Japanese (Jap), Chinese (Man), Cantonese (Can) and English (Eng) origins. D_C consists of personal names of Japanese (Jap), Chinese (Chi, including both Mandarin and Cantonese) and English (Eng) origins. Table 1 and Table 2 list their details. In the experiments, 90% of entries in Table 1 (D_E) and Table 2 (D_C) are randomly selected for training and the remaining 10% are kept for testing for each language origin. Columns 2 and 3 in Tables 7 and 8 list the numbers of entries in the training and test sets.

Evaluation Methods: Accuracy is usually used to evaluate the recognition performance (Qu and Gregory, 2004; Li *et al.*, 2006; Li *et al.*, 2007). However, as we know, the individual accuracy used before only reflects the performance of *recall* and does not give a whole picture about a multi-class classification task. Instead, we use *precision* (*P*), *recall* (*R*) and F-measure (*F*) to evaluate the performance of each origin. In addition, an overall accuracy (*Acc*) is also given to describe the whole performance. The *P*, *R*, *F* and *Acc* are calculated as following:

$$P = \frac{\#\ correctly\ recognized\ entries\ of\ the\ given\ origin}{\#\ entries\ recognized\ as\ the\ given\ origin\ by\ the\ system}$$

$$R = \frac{\#\ correctly\ recognized\ entries\ of\ the\ given\ origin}{\#\ entries\ of\ the\ given\ origin}$$

$$F = \frac{2PR}{P+R} \qquad Acc = \frac{\#\ all\ correctly\ recognized\ entries}{\#\ all\ entries}$$

4.2 Experimental Results and Analysis

Table 3 reports the experimental results of ENOR. It shows that the MaxEnt approach achieves the best result of 98.44% in overall accuracy when combining all the diverse features as listed in Subsection 3.2. Table 3 also measures the contributions of different features for ENOR by gradually incorporating the feature set. It shows that:

1) All individual features are useful since the performance increases consistently when more features are being introduced.
2) Bigram feature presents the most informative feature that gives rise to the highest performance gain, while the trigram feature further boosts performance too.
3) MaxEnt method can integrate the advantages of previous rule-based and statistical methods and easily integrate other features.

Features	Origin	*P(%)*	*R(%)*	*F*	*Acc(%)*
FUni	Eng	91.40	80.76	85.75	85.29
	Man	83.05	81.90	82.47	
	Can	81.13	82.76	81.94	
	Jap	87.31	94.11	90.58	
+FBi	Eng	97.54	91.10	94.21	96.72
	Man	97.51	98.10	97.81	
	Can	97.68	98.05	97.86	
	Jap	94.62	98.24	96.39	
+FTri	Eng	97.71	93.79	95.71	97.97
	Man	98.94	99.37	99.16	
	Can	99.12	99.19	99.15	
	Jap	96.19	98.52	97.34	
+FPUni	Eng	97.53	94.64	96.06	98.16
	Man	99.21	99.43	99.32	
	Can	99.41	99.24	99.33	
	Jap	96.48	98.49	97.47	
+FPBi	Eng	97.68	94.98	96.31	98.28
	Man	99.32	99.50	99.41	
	Can	99.53	99.34	99.44	
	Jap	96.59	98.52	97.55	
+FPTri	Eng	97.62	94.97	96.27	98.30
	Man	99.34	99.58	99.46	
	Can	99.63	99.37	99.50	
	Jap	96.61	98.45	97.52	
+FFre	Eng	97.74	95.06	96.38	98.35
	Man	99.37	99.59	99.48	
	Can	99.61	99.41	99.51	
	Jap	96.66	98.56	97.60	
+ FMan + FCan	Eng	97.82	95.11	96.45	98.44
	Man	99.52	99.68	99.60	
	Can	99.71	99.59	99.65	
	Jap	96.69	98.59	97.63	

Table 3: Contribution of each feature for ENOR

Features	Eng	Jap	Man	Can
FMan	-0.357	0.069	0.072	-0.709
FCan	-0.424	-0.062	-0.775	0.066

Table 4: Features weights in ENOR task.

Feature	Origin	*P*(%)	*R*(%)	*F*	*Acc*(%)
FUni	Eng	97.89	98.43	98.16	96.97
	Chi	95.80	95.03	95.42	
	Jap	96.96	97.05	97.00	
+FBi	Eng	96.99	98.27	97.63	96.28
	Chi	96.86	92.11	94.43	
	Jap	95.04	97.73	96.36	
+FLen	Eng	97.35	98.38	97.86	97.14
	Chi	97.29	95.00	96.13	
	Jap	96.78	97.64	97.21	
+FPUni	Eng	97.74	98.65	98.19	97.77
	Chi	97.65	96.34	96.99	
	Jap	97.91	98.05	97.98	
+FPBi	Eng	97.50	98.43	97.96	97.56
	Chi	97.61	96.04	96.82	
	Jap	97.59	97.94	97.76	
FUni +FLen + FPUni	Eng	98.08	99.04	98.56	98.10
	Chi	97.57	96.88	97.22	
	Jap	98.58	98.11	98.34	

Table 5: Contribution of each feature for CNOR

Table 4 reports the feature weights of two features "FMan" and "FCan" with regard to different origins in ENOR task. It shows that "FCan" has positive weight only for origin "Can" while "FMan" has positive weights for both origins "Man" and "Jap", although the weight for "Man" is higher. This agrees with our observation that the two features favor origins "Man" or "Can". The feature weights also reflect the fact that some Japanese names can be successfully parsed by the Chinese Mandarin *Pinyin* system due to their similar syllable structure. For example, the Japanese name "Tanaka Miho" is also a sequence of Chinese *Pinyin*: "Ta-na-ka Mi-ho".

Table 5 reports the contributions of different features in CNOR task by gradually incorporating the feature set. It shows that:

1) Unigram features are the most informative
2) Bigram features degrade performance. This is largely due to the data sparseness problem as discussed in Section 3.2.
3) FLen is also useful that confirms our intuition about name length.

Finally the combination of the above three useful features achieves the best performance of 98.10% in overall accuracy for CNOR as in the last row of Table 5.

In Tables 3 and 5, the effectiveness of each feature may be affected by the order in which the features are incorporated, i.e., the features that are added at a later stage may be underestimated. Thus, we conduct another experiment using "all-but-one" strategy to further examine the effectiveness of each kind of features. Each time, one type of the n-gram (n=1, 2, 3) features (including orthographic n-gram, position-specific and n-gram frequency features) is removed from the whole feature set. The results are shown in Table 6.

Features	Origin	*P*(%)	*R*(%)	*F*	*Acc*(%)
w/o Unigram	Eng	97.81	95.01	96.39	98.34
	Man	99.41	99.58	99.49	
	Can	99.53	99.48	99.50	
	Jap	96.63	98.52	97.57	
w/o Bigram	Eng	97.34	95.17	96.24	98.26
	Man	99.30	99.48	99.39	
	Can	99.54	99.33	99.43	
	Jap	96.73	98.32	97.52	
w/o Trigram	Eng	97.57	94.10	95.80	97.94
	Man	98.98	99.23	99.10	
	Can	99.20	99.08	99.14	
	Jap	96.06	98.42	97.23	

Table 6: Effect of n-gram feature for ENOR

Table 6 reveals that removing trigram features affects the performance most. This suggests that trigram features are much more effective for ENOR than other two types of features. It also shows that trigram features in ENOR does not suffer from the data sparseness issue.

As observed in Table 5, in CNOR task, 93.96%

accuracy is obtained when removing unigram features, which is much lower than 98.10% when bigram features are removed. This suggests that unigram features are very useful in CNOR, which is mainly due to the data sparseness problem that bigram features may have encountered.

4.3 Model Complexity and Data Sparseness

Table 7 (ENOR) and Table 8 (CNOR) compare our MaxEnt model with the SUM model (Qu and Gregory, 2004) and the PP model (Li *et al.*, 2006). All the experiments are conducted on the same data sets as described in section 4.1. Tables 7 and 8 show that the proposed MaxEnt model outperforms other models. The results are statistically significant (χ^2 test with $p<0.01$) and consistent across all tests.

Model Complexity:

We look into the complexity of the models and their effects. Tables 7 and 8 summarize the overall accuracy of three models. Table 9 reports the numbers of parameters in each of the models. We are especially interested in a comparison between the MaxEnt and PP models because their performance is close. We observe that, using trigram features, the MaxEnt model has many more parameters than the PP model does. Therefore, it is not surprising if the MaxEnt model outperforms when more training data are available. However, the experiment results also show that the MaxEnt model consistently outperforms the PP model even with the same size of training data. This is largely attributed to the fact that MaxEnt incorporates more robust features than the PP model does, such as rule-based, length of names features.

One also notices that PP clearly outperforms SUM by using the same number of parameters in ENOR and shows comparable performance in CNOR tasks. Note that SUM and PP are different in two areas: one is the PP model employs word length normalization while SUM doesn't; another that the PP model uses n-gram conditional probability while SUM uses n-character joint probability. We believe that the improved performance of PP model can be attributed to the effect of usage of conditional probability, rather than length normalization since length normalization does not change the order of probabilities.

Data Sparesness:

We understand that we can only assess the effectiveness of a feature when sufficient statistics is available. In CNOR (see Table 8), we note that the Chinese transliterations of English origin use only 377 Chinese characters, so data sparseness is not a big issue. Therefore, bigram SUM and bigram PP methods easily achieve good performance for English origin. However, for Japanese origin (represented by 1413 Chinese characters) and Chinese origin (represented by 2319 Chinese characters), the data sparseness becomes acute and causes performance degradation in SUM and PP models. We are glad to find that MaxEnt still maintains a good performance benefiting from other robust features.

Table 10 compares the overall accuracy of the three methods using unigram and bigram features in CNOR task, respectively. It shows that the MaxEnt method achieves best performance. Another interesting finding is that unigram features perform better than bigram features for PP and MaxEnt models, which shows that data sparseness remains an issue even for MaxEnt model.

5 Conclusion

We propose using MaxEnt model to explore diverse features for name origin recognition. Experiment results show that our method is more effective than previously reported methods. Our contributions include:

1) Cast the name origin recognition problem as a multi-class classification task and propose a MaxEnt solution to it;
2) Explore and integrate diverse features for name origin recognition and propose the most effective feature sets for ENOR and for CNOR

In the future, we hope to integrate our name origin recognition method with a machine transliteration engine to further improve transliteration performance. We also hope to study the issue of name origin recognition in context of sentence and use contextual words as additional features.

References

Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics. 22(1):39–71.

William B. Cavnar and John M. Trenkle. 1994. Ngram based text categorization. In 3rd Annual Symposium

on Document Analysis and Information Retrieval, 275–282.

Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. Computational Linguistics. 24(4), 599-612.

Jin-Shea Kuo and Ying-Kuei Yan. 2004. Generating Paired Transliterated-Cognates Using Multiple Pronunciation Characteristics from Web Corpora. PACLIC 18, December 8th-10th, Waseda University, Tokyo, Japan, 275–282.

Haizhou Li, Shuanhu Bai and Jin-Shea Kuo. 2006. Transliteration. Advances in Chinese Spoken Language Processing. World Scientific Publishing Company, USA, 341–364.

Haizhou Li, Khe Chai Sim, Jin-Shea Kuo and Minghui Dong. 2007. Semantic Transliteration of Personal Names. ACL-2007. 120–127.

Xinhua News Agency. 1992. Chinese Transliteration of Foreign Personal Names. The Commercial Press

Yan Qu and Gregory Grefenstette. 2004. Finding ideographic representations of Japanese names written in Latin script via language identification and corpus validation. ACL-2004. 183–190.

Min Zhang, Jian Su and Haizhou Li. 2004. Direct Orthographical Mapping for Machine Translation. COLING-2004. 716-722.

Origin	**# training entries**	**# test entries**	**Trigram SUM**			**Trigram PP**			**MaxEnt**		
			***P*(%)**	***R*(%)**	***F***	***P*(%)**	***R*(%)**	***F***	***P*(%)**	***R*(%)**	***F***
Eng	79,920	8,879	94.66	72.50	82.11	95.84	94.72	95.28	97.82	95.11	**96.45**
Man	104,291	11,588	86.79	94.87	90.65	98.99	98.33	98.66	99.52	99.68	**99.60**
Can	104,165	11,574	90.03	93.87	91.91	96.17	99.67	97.89	99.71	99.59	**99.65**
Jap	110,951	12,324	89.17	92.84	90.96	98.20	96.29	97.24	96.69	98.59	**97.63**
Overall *Acc* (%)			89.57			97.39			**98.44**		

Table 7: Benchmarking different methods in ENOR task

Origin	**# training entries**	**# test entries**	**Bigram SUM**			**Bigram PP**			**MaxEnt**		
			***P*(%)**	***R*(%)**	***F***	***P*(%)**	***R*(%)**	***F***	***P*(%)**	***R*(%)**	***F***
Eng	37,644	3,765	95.94	98.65	97.28	97.58	97.61	97.60	98.08	99.04	**98.56**
Chi	29,795	2,980	96.26	87.35	91.59	95.10	87.35	91.06	97.57	96.88	**97.22**
Jap	33,897	3,390	93.01	97.67	95.28	90.94	97.43	94.07	98.58	98.11	**98.34**
Overall *Acc* (%)			95.00			94.53			**98.10**		

Table 8: Benchmarking different methods in CNOR task

Methods	**# of parameters for ENOR**	**# of parameters for CNOR**	
	Trigram	**Unigram**	**Bigram**
MaxEnt	124,692	13,496	182,116
PP	16,851	4,045	86,490
SUM	16,851	4,045	86,490

Table 9: Numbers of parameters used in different methods

	SUM	**PP**	**MaxEnt**
Unigram Features	90.55	97.09	**98.10**
Bigram Features	95.00	94.53	**97.56**

Table 10: Overall accuracy using unigram and bigram features in CNOR task

A More Discerning and Adaptable Multilingual Transliteration Mechanism for Indian Languages

Harshit Surana
Language Tech. Research Centre
IIIT, Hyderabad, India
surana.h@gmail.com

Anil Kumar Singh
Language Tech. Research Centre
IIIT, Hyderabad, India
anil@research.iiit.ac.in

Abstract

Transliteration is the process of transcribing words from a source script to a target script. These words can be content words or proper nouns. They may be of local or foreign origin. In this paper we present a more discerning method which applies different techniques based on the word origin. The techniques used also take into account the properties of the scripts. Our approach does not require training data on the target side, while it uses more sophisticated techniques on the source side. Fuzzy string matching is used to compensate for lack of training on the target side. We have evaluated on two Indian languages and have achieved substantially better results (increase of up to 0.44 in MRR) than the baseline and comparable to the state of the art. Our experiments clearly show that word origin is an important factor in achieving higher accuracy in transliteration.

1 Introduction

Transliteration is a crucial factor in Cross Lingual Information Retrieval (CLIR). It is also important for Machine Translation (MT), especially when the languages do not use the same scripts. It is the process of transforming a word written in a source language into a word in a target language without the aid of a resource like a bilingual dictionary. Word pronunciation is usually preserved or is modified according to the way the word should be pronounced in the target language. In simple terms, it means finding out how a source word should be written in the script of the target languages such that it is acceptable to the readers of the target language.

One of the main reasons of the importance of transliteration from the point of view of Natural Language Processing (NLP) is that Out Of Vocabulary (OOV) words are quite common since every lexical resource is very limited in practical terms. Such words include named entities, technical terms, rarely used or 'difficult' words and other borrowed words, etc. The OOV words present a challenge to NLP applications like CLIR and MT. In fact, for very close languages which use different scripts (like Hindi and Urdu), the problem of MT is almost an extension of transliteration.

A substantial percentage of these OOV words are named entities (AbdulJaleel and Larkey, 2003; Davis and Ogden, 1998). It has also been shown that cross language retrieval performance (average precision) reduced by more than 50% when named entities in the queries were not transliterated (Larkey et al., 2003).

Another emerging application of transliteration (especially in the Indian context) is for building input methods which use QWERTY keyboard for people who are more comfortable typing in English. The idea is that the user types Roman letters but the input method transforms them into letters of Indian language (IL) scripts. This is not as simple as it seems because there is no clear mapping between Roman letters and IL letters. Moreover, the output word should be a valid word. Several commercial efforts have been started in this direction due to the lack of a good (and familiar) input mech-

anism for ILs. These efforts include the Google Transliteration mechanism[1] and Quilpad[2]. (Rathod and Joshi, 2002) have also developed more intuitive input mechanisms for phonetic scripts like Devanagari.

Our efforts take into account the type of the word, the similarities among ILs and the characteristics of the Latin and IL scripts. We use a sophisticated technique and machine learning on the source language (English) side, while a simple and light technique on the target (IL) side. The advantage of our approach is that it requires no resources except unannotated corpus (or pages crawled from the Web) on the IL side (which is where the resources are scarce). The method easily generalizes to ILs which use Brahmi origin scripts. Our method has been designed such that it can be used for more conventional applications (MT, CLIR) as well as for applications like building an input mechanism.

Much of the work for transliteration in ILs has been done from one Indian script to another. One of the major work is of Punjabi machine transliteration (Malik, 2006). This work tries to address the problem of transliteration for Punjabi language from Shahmukhi (Arabic script) to Gurmukhi using a set of transliteration rules (character mappings and dependency rules). *Om* transliteration scheme (Ganapathiraju et al., 2005) also provides a script representation which is common for all Indian languages. The display and input are in human readable Roman script. Transliteration is partly phonetic. (Sinha, 2001) had used Hindi Transliteration used to handle unknowns in MT.

naukri (A popular domain name)	722,000
nokri (domain name)	19,800
naukari	10,500
naukary (domain name)	5,490
nokari	665
naukarii	133
naukaree	102

Table 1: Variations of a Hindi Word nOkarI (job). The numbers are pages returned when searching on Google.

[1]www.google.co.in/press/pressrel/news_transliteration.html
[2]www.quillpad.com

Aswani et. al (Aswani and Gaizauskas, 2005) have used a transliteration similarity mechanism to align English-Hindi parallel texts. They used character based direct correspondences between Hindi and English to produce possible transliterations. Then they apply edit distance based similarity to select the most probable transliteration in the English text. However, such method can only be appropriate for aligning parallel texts as the number of possible candidates is quite small.

The paper is structured as follows. In Section-2, we discuss the problem of a high degree of variation in Indian words, especially when written in Latin script. In Section-3, we explain the idea of using information about the word origin for improving transliteration. Then in Section-4 we describe the method that we use for guessing the word origin. Once the word origin is guessed, we can apply one of the two methods for transliteration depending on the word origin. These two methods are described in Section-5 and Section-6, respectively. Fuzzy string matching, which plays an important role in our approach, is described in Section-7. In Section-8 we put together all the elements covered in the preceding sections and explain the Discerning Adaptable Transliteration Mechanism. Section-9 presents the evaluation of our approach in comparison with two baseline methods, one of which uses knowledge about word origin. Finally, in Section-10 we present the conclusions.

2 Variation in Indian Words in Latin Script

Since the purpose of our work is not only to transliterate named entities but to be useful for applications like input mechanisms, we had to consider some other issues too which may not be considered directly related to transliteration. One of these is that there is a lot of spelling variation in ILs. This variation is much more when the IL words are written using the Latin script (Table-1). In other words, the amount of ambiguity is very high when we try to build a system that can be used for purposes like designing input mechanisms, instead of just for transliteration of NEs etc. for MT or CLIR. One reason for very high variation in the latter case is that unlike Romaji for Japanese (which is taught in

schools in Japan), there is no *widely adopted* transliteration scheme using the Latin script, although there are a number of standard schemes, which are not used by common users. At present the situation is that most Indians use Indian scripts while writing in ILs, but use the Latin script when communicating online. ILs are rarely used for official communication, except in government offices in some states.

3 Word Origin and Two Ways of Transliteration

Previous work for other languages has shown that word origin plays a part in how the word should be transliterated(Oh and Choi, 2002; May et al., 2004). Llitjos and Black (Llitjos and Black, 2001) had shown that the knowledge of language origin can substantially improve pronunciation generation accuracy. This information has been used to get better results (Oh and Choi, 2002). They first checked whether the word origin is Greek or not before selecting one of the two methods for transliteration. This approach improved the results substantially. However, they had used a set of prefixes and suffixes to identify the word origin. Such an approach is not scalable. In fact, in a large number of cases, word origin cannot be identified by using list of affixes.

For ILs, we also define two categories of words: words which can be roughly considered Indian and those which can be roughly considered foreign. Note that 'Indian' and 'foreign' are just loose labels here. Indian words, which include proper nouns and also common vocabulary words, are more relevant in applications like input methods. Two different methods are used for transliterating, as explained later.

4 Disambiguating Word Origin

Previously (Llitjos and Black, 2001) used probabilities of all trigrams to belong to a particular language as an measure to disambiguate word origins. We use a more sophisticated method that has been successfully used for language and encoding identification (Singh, 2006a).

We first prepare letter based 5-gram models from the lists of two kinds of words (Indian and foreign). Then we combine n-grams of all orders and rank them according to their probability in descending order. Only the top N n-grams are retained and the rest are pruned. Now we have two probability distributions which can be compared by a measure of distributional similarity. The measure used is symmetric cross entropy or SCE (Singh, 2006a).

Since the accuracy of identification is low if test data is very low, which is true in our case because we are trying to identify the class of a single word, we had to extend the method used by Singh. One major extension was that we add word beginning and ending markers to all the words in training as well as test data. This is because n-grams at beginning, middle and end of words should be treated differently if we want to identify the 'language' (or class) of the word.

For every given word, we get a probability about its origin based on SCE. Based on this probability measure, transliteration is performed using different techniques for different classes (Indian or foreign). In case of ambiguity, transliteration is performed using both methods and the probabilities are used to get the final ranking of all possible transliterations.

5 Transliteration of Foreign Words

These words include named entities (George Bush) and more common nouns (station, computer) which are regularly used in ILs. To generate transliteration candidates for such words, we first try to guess the word pronunciation or use a lookup dictionary (if available) to find it. Then we use some simple manually created mappings, which can be used for all Indian languages. Note that these mappings are very few in number (Figure-1 and Figure-2) and can be easily created by non-linguistically trained people. They play only a small role in the method because other steps (like fuzzy string matching) do most of the work.

For our experiments, we used the CMU speech dictionary as the lookup, and also to train pronunciation estimation. If a word is not in the CMU dictionary, we estimate the word pronunciation, as explained later.

We directly map from English phonemes to IL letters. This is based on our observation that a foreign word is usually transliterated in almost the same way as it is pronounced. Almost all English phonemes can be roughly mapped to specific letters (representing phonemes, as IL scripts are phonetic in na-

ture) in ILs. Similar observations have been made about Hindi by Su-Youn Yoon, Kyoung-Young Kim and Richard Sproat (Yoon et al., 2007). We have prepared our own mappings with help from native speakers of the languages concerned, which is relatively quite a simple task since the letters in Indic scripts correspond closely with phonemes.

6 Transliteration of Indian Words

These words include (mainly Indian) named entities of (e.g. Taj Mahal, Manmohan Singh) and common vocabulary words (common nouns, verbs) which need to be transliterated. They also include words which are spelled similar to the way Indian words are spelled when written in Latin (e.g. Baghdad, Husain). As stated earlier, this class of words are much more relevant for an input method using a QWERTY keyboard.

Since words of Indian origin usually have phonetic spellings when they are written in English (Latin), the issue of pronunciation estimation or lookup is not important. However, there can be many possible vowel and consonant segments which can be formed out of a single word. For example 'ai' can be interpreted as a single vowel with sound AE (as in Husain), or as two vowels AA IH (as in Rai). To perform segmentation, we have a simple program which produces candidates for all possible segments. This program uses a few rules defining the possible consonant and vowel combinations.

Now we simply map these segments to their nearest IL letters (or letter combinations). This is also done using a simple set of mappings, which do not contain any probabilities or contexts. This step generates transliteration candidates. These are then filtered and ranked using fuzzy string matching.

7 Fuzzy String Matching

The initial steps use simpler methods to generate transliteration candidates on the source as well as the target side. They also use no resources on the target (IL) side. The step of fuzzy string matching compensates for the lack of more language specific knowledge during the earlier phase. The transliteration candidates are matched with the words in the target language corpus (actually, words in the word list extracted from the corpus). The fuzzy string matching algorithm we use is finely tuned for Indian Languages and performs much better than language independent approaches like edit distance (Singh et al., 2007). This method can be used for all the languages which use Abugida scripts, e.g. Hindi, Bengali, Telugu, Amharic, Thai etc. It uses characteristics of a writing system for fuzzy search and is able to take care of spelling variation, which is very common in these languages. This method shows an improvement in F-measure of up to 30% over scaled edit distance.

AA	आ|औ	ఆ । ఔ
B	ब	బ
CH	च	చ
D	ड	డ
DH	थ	ధ
F	फ	ఫ
JH	ज	జ
L	ल	ల
M	म	మ
NG	न्ग|न्क	న్గ । న్క
P	प	ప
TH	थ	ధ
UH	उ	ఉ
ZH	स|ज	స । జ

Figure 1: Mappings for foreign words. The three columns are for Roman, Devanagari and Telugu

The method for fuzzy string matching is based on the Computational Phonetic Model of Scripts or CPMS (Singh, 2006b), which models scripts (specifically Indic scripts) in terms of phonetic (articulatory) and orthographic features. For calculating the distance between two letters it uses a Stepped Distance Function (SDF). Each letter is represented as a vector of features. Then, to calculate the distance between two strings, it uses an adapted version of the Dynamic Time Warping algorithm (My-

A	अ	అ
AA	आ	ఆ
BH	भ	భ
CH	छ|च	ఛ। చ
D	ड|द	డ। ద
E	ऐ	ఐ
F	फ	ఫ
L	ल	ల
M	म	మ
N	न|ण	న। ణ
OO	ऊ	ఊ
R	र	ర
S	स	స
Z	ज	జ

Figure 2: Mappings for Indian Words

ers, 1980). In the fuzzy string matching method that we use (Singh et al., 2007), an *akshar* (roughly a syllable) is used as the unit, instead of a letter.

8 Discerning Adaptable Transliteration Mechanism (DATM)

We use the above mentioned steps to transliterate a given word based on its origin. In case of ambiguity of word origin both methods are used, and possible transliterations are ranked. Based on the class of the word, the possible pronunciations (for foreign words) and the possible segmentations (for Indian words) are generated. Then, for foreign words, English phonemes are mapped to IL segments. For Indian words, Latin segments are mapped to IL segments.

Now, the transliteration candidates are matched with target language words, using the fuzzy text search method (Singh et al., 2007). Possible transliterations are ranked based on three parameters: word frequency, text search cost and the probability of the word belonging to the class through which it is transliterated. A block diagram describing the method is shown in Figure-3. The ranks are obtained on the basis of a score which is calculated using the following formula:

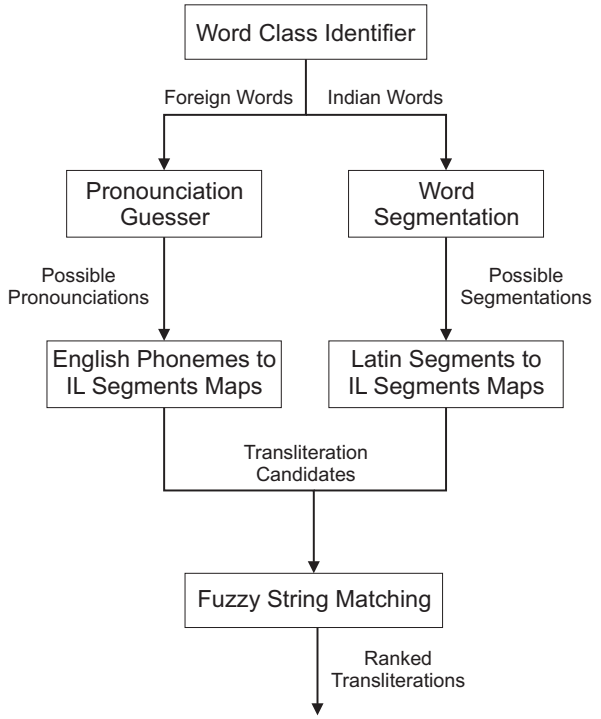


Figure 3: Block Diagram of the Discerning Adaptive Transliteration Method (DATM)

$$T_t = \frac{log(f_t) * p(C \mid s)}{cost(c,\ t) + K} \quad (1)$$

where T_t is the transliteration score for the target word t, f_t is the frequency of t in the target language corpus, C is the word class (foreign or Indian), s is the source word, c is a transliteration candidate which has been generated depending on the predicted class C, $p(C|s)$ is the probability of the class C given s, $cost(c,\ t)$ is the cost of fuzzy string matching between c and t, and finally K is a constant which determines how much weight is given to the cost of fuzzy string matching.

9 Evaluation

We evaluate our method for two major languages of India: Hindi and Telugu. We compare our results with a very commonly used method (Oh and Choi, 2006) based on bilingual dictionary to learn translit-

Language →	English-Hindi		English-Telugu	
Method ↓	**MRR**	**Pr**	**MRR**	**Pr**
DATM	0.87	80%	0.82	71%
DBL	0.56	47%	0.53	46%
BL	0.43	35%	0.43	37%

DATM: Discerning Adaptive Transliteration Mechanism
DBL: Discerning Baseline Method
BL: Baseline Method

MRR: Mean Reciprocal Rank
Pr: Precision

Table 2: Evaluation on English-Hindi and English-Telugu

erations. As there are no bilingual transliteration dictionaries available for ILs, we had to create our own resources.

9.1 Experimental Setup

We created 2000-word lists which consisted of both foreign and Indian words written in Latin script and their transliterations in Hindi and Telugu. This dictionary was created by people with professional knowledge in both English and the respective Indian language. We only use this list for training the baseline method, as our method does not need training data on the target side. The size of bilingual word lists that we are using is less than those used for experiments by some other researchers. But our approach focuses on developing transliterations for languages with resource scarcity. This setup is more meaningful for languages with scarce resources.

Since, normal transliteration mechanisms do not consider word origin, we train the baseline using the set of 2000 words containing both foreign and Indian words. Alignments from English to respective Indian languages were learned by aligning these lists using GIZA++. The alignments obtained were fed into a maximum entropy classifier with a context window size of 2 (3 is generally considered better window size, but because the training size is not huge, a context window of 3 gave substantially worse results). This method is similar to the grapheme based model as described by Oh and Choi (Oh and Choi, 2006). However, unlike in their approach, the candidate pairs are matched with words in the target language and are ranked based on edit distance (BL).

For our method (DATM), we have used CMU dictionary and a collection of Indian named entities (written in Latin) extracted from web to train the language identification module. We have considered n-grams of order 5 and pruned them by 3500 frequency. In case the foreign word is not found in CMU Speech dictionary, we guess its pronunciation using the method described by Oh and Choi. However, in this case, the context window size is 3.

We also use another method (DBL) to check the validity of our assumptions about word origin. We use the same technique as BL, but in this case we train two models of 1000 words each, foreign and Indian. To disambiguate which model to use, we use the same language identification method as in DATM.

9.2 Results

To evaluate our method we have created word lists of size 200 which were doubly checked by two individuals. These also contain both Indian and Foreign words. We use both precision and mean reciprocal rank (MRR) to evaluate our method against baseline (BL) and discerning baseline (DBL). MRR is a measure commonly used in information retrieval when there is precisely one correct answer (Kandor and Vorhees, 2000). Results can be seen in Table-2. The highest scores were obtained for Hindi using DATM. The MRR in this case was 0.87.

One important fact that comes out from the results is that determining the class of a word and then using an appropriate method can lead to significant increase in performance. This is clear from the results for BL and DBL. The only difference between

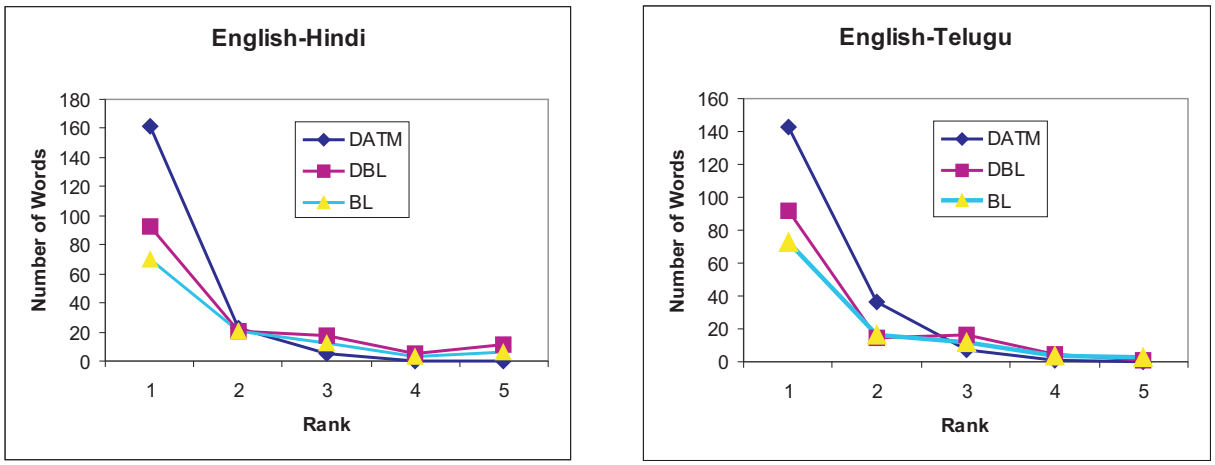


Figure 4: Number of Correct Words vs. Rank. A significantly higher percentage of correct words occur at rank 1 for the DATM method, as compared to BL and DBL methods. This percentage indicates a more practical view of the accuracy transliteration algorithm.

these two was that two different models were trained for the two classes. Then the class of the word was identified (in DBL) and the model trained for that class was used for transliteration.

It should be noted that Yoon et al. (Yoon et al., 2007) have also reported MRR score on Hindi. They have used a number of phonetic and pseudo features, and trained their algorithm on a winnow classifier. They tested their algorithm only for named entities. They have considered a relatively limited number of candidate words on the target language side (1,500) which leads to 150k pairs on which they have evaluated their method. They have reported the results as 0.91 and 0.89 under different test conditions. In case of our evaluation, we do not restrict the candidate words on the target side except that it should be available in the corpus. Because of this formulation, there are over 1000k words for Hindi and over 1800k words from Telugu. This leads to a extremely high number of pairs possible. But such an approach is also necessary as we want our algorithm to be scalable to bigger sizes and also because there are no high quality tools (like named entity recognizers) for Indian languages. This is one of the reason for relatively (compared to figures reported by other researchers) low baseline scores. Despite all these issues, our simpler approach yields similar results.

Figure-4 shows how the number of correct words varies with the rank.

Two possible issues are the out of vocabulary (OOV) words and misspelled or foreign words in the IL corpus. The OOV words are not handled right now by our method, but we plan to extend our method to at least partially take care of such words. The second issue is mostly resolved by our use of fuzzy string matching, although there is scope for improvement.

10 Conclusions and Further Work

We presented a more general and adaptable method for transliteration which is especially suitable for Indian languages. This method first identifies the class (foreign or Indian) of the word on the source side. Based on the class, one of the two methods is used for transliteration. Easily creatable mapping tables and a fuzzy string matching algorithm are then used to get the target word. Our evaluations shows that the method performs substantially better than the two baselines we tested against. The results are better in terms of both MRR (up to 0.44) and precision (45%). Our method is designed to be used for other applications like tolerant input methods for Indian languages and it uses no resources on the target languages side except an unannotated corpus. The results can be further improved if we consider context information too.

We have also shown that disambiguating word origin and applying an appropriate method could be

critical in getting good transliterations. Currently we are assuming that the word to be transliterated is in the target language corpus. We plan to extend the method so that even those words can be transliterated which are not in the target language corpus. We are also working on using this method for building a tolerant input method for Indian languages and on integrating the transliteration mechanism as well as the input method with an open source NLP friendly editor called Sanchay Editor (Singh, 2008).

References

N. AbdulJaleel and L.S. Larkey. 2003. Statistical transliteration for english-arabic cross language information retrieval. *Proceedings of the twelfth international conference on Information and knowledge management*, pages 139–146.

N. Aswani and R. Gaizauskas. 2005. A hybrid approach to align sentences and words in English-Hindi parallel corpora. *Proceedings of the ACL Workshop on" Building and Exploiting Parallel Texts.*

M.W. Davis and W.C. Ogden. 1998. Free resources and advanced alignment for cross-language text retrieval. *Proceedings of the 6th Text Retrieval Conference (TREC-6)*, pages 385–402.

M. Ganapathiraju, M. Balakrishnan, N. Balakrishnan, and R. Reddy. 2005. OM: One Tool for Many (Indian) Languages. *ICUDL: International Conference on Universal Digital Library, Hangzhou*.

L. Larkey, N. AbdulJaleel, and M. Connell. 2003. What's in a Name? Proper Names in Arabic Cross-Language Information Retrieval. Technical report, CIIR Technical Report, IR-278.

A. Llitjos and A. Black. 2001. Knowledge of language origin improves pronunciation of proper names. *Proceedings of EuroSpeech-01*, pages 1919–1922.

M.G.A. Malik. 2006. Punjabi Machine Transliteration. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 1137–1144.

J. May, A. Brunstein, P. Natarajan, and R. Weischedel. 2004. Surprise! What's in a Cebuano or Hindi Name? *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):169–180.

C. S. Myers. 1980. *A Comparative Performance Study of Several Dynamic Time Warping Algorithms for Speech Recognition*. Ph.D. thesis, M.I.T., Cambridge, MA, Feb. http://gate.ac.uk.

J.H. Oh and K.S. Choi. 2002. An English-Korean transliteration model using pronunciation and contextual rules. *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7.

J.H. Oh and K.S. Choi. 2006. An ensemble of transliteration models for information retrieval. *Information Processing and Management: an International Journal*, 42(4):980–1002.

A. Rathod and A. Joshi. 2002. A Dynamic Text Input scheme for phonetic scripts like Devanagari. *Proceedings of Development by Design (DYD)*.

Anil Kumar Singh, Harshit Surana, and Karthik Gali. 2007. More accurate fuzzy text search for languages using abugida scripts. In *Proceedings of ACM SIGIR Workshop on Improving Web Retrieval for Non-English Queries*, Amsterdam, Netherlands.

Anil Kumar Singh. 2006a. Study of some distance measures for language and encoding identification. In *Proceedings of ACL 2006 Workshop on Linguistic Distance*, Sydney, Australia.

Anil Kumar Singh. 2006b. A computational phonetic model for indian language scripts. In *Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*, Nijmegen, The Netherlands.

Anil Kumar Singh. 2008. A mechanism to provide language-encoding support and an nlp friendly editor. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, Hyderabad, India.

RMK Sinha. 2001. Dealing with unknowns in machine translation. *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, 2.

S.Y. Yoon, K.Y. Kim, and R. Sproat. 2007. Multilingual Transliteration Using Feature based Phonetic Method. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 112–119.

UCSG: A Wide Coverage Shallow Parsing System

G. Bharadwaja Kumar and **Kavi Narayana Murthy**
Department of Computer and Information Sciences
University of Hyderabad
g_vijayabharadwaj@yahoo.com, knmuh@yahoo.com

Abstract

In this paper, we propose an architecture, called UCSG Shallow Parsing Architecture, for building wide coverage shallow parsers by using a judicious combination of linguistic and statistical techniques without need for large amount of parsed training corpus to start with. We only need a large POS tagged corpus. A parsed corpus can be developed using the architecture with minimal manual effort, and such a corpus can be used for evaluation as also for performance improvement. The UCSG architecture is designed to be extended into a full parsing system but the current work is limited to chunking and obtaining appropriate chunk sequences for a given sentence. In the UCSG architecture, a Finite State Grammar is designed to accept *all* possible chunks, referred to as word groups here. A separate statistical component, encoded in HMMs (Hidden Markov Model), has been used to rate and rank the word groups so produced. Note that we are not pruning, we are only rating and ranking the word groups already obtained. Then we use a Best First Search strategy to produce parse outputs in best first order, without compromising on the ability to produce all possible parses in principle. We propose a bootstrapping strategy for improving HMM parameters and hence the performance of the parser as a whole.

A wide coverage shallow parser has been implemented for English starting from the British National Corpus, a nearly 100 Million word POS tagged corpus. Note that the corpus is not a parsed corpus. Also, there are tagging errors, multiple tags assigned in many cases, and some words have not been tagged. A dictionary of 138,000 words with frequency counts for each word in each tag has been built. Extensive experiments have been carried out to evaluate the performance of the various modules. We work with large data sets and performance obtained is encouraging. A manually checked parsed corpus of 4000 sentences has also been developed and used to improve the parsing performance further. The entire system has been implemented in Perl under Linux.

Key Words:- Chunking, Shallow Parsing, Finite State Grammar, HMM, Best First Search

1 Introduction

In recent times, there has been an increasing interest in wide coverage and robust but shallow parsing systems. Shallow parsing is the task of recovering only a limited amount of syntactic information from natural language sentences. Often shallow parsing is restricted to finding phrases in sentences, in which case it is also called chunking. Steve Abney (Abney, 1991) has described chunking as *finding syntactically related non-overlapping groups of words.* In CoNLL chunking task (Tjong Kim Sang and Buchholz, 2000) chunking was defined as *the task of dividing a text into syntactically non-overlapping phrases.*

Most of the shallow parsers and chunkers described in literature (Tjong Kim Sang and Buchholz, 2000; Carreras and Marquez, 2003; Dejean, 2002; Molina and Pla, 2002; Osborne, 2002; Sang, 2002; Abney, 1996; Grefenstette, 1996; Roche, 1997) have used either only rule based techniques or only machine learning techniques. Hand-crafting rules in the linguistic approach can be very laborious and time consuming. Parsers tend to produce a large number of possible parse outputs and in the absence

of suitable rating and ranking mechanisms, selecting the right parse can be very difficult. Statistical learning systems, on the other hand, require large and representative parsed corpora for training, and such training corpora are not always available. Perhaps only a good combination of linguistic and statistical approaches can give us the best results with minimal effort.

Other important observations from literature that motivated the present work are: 1) Most chunking systems have so far been tested only on small scale data 2) Good performance has been obtained only under restricted conditions 3) Performance is often evaluated in terms of individual chunks rather than complete chunk sequences for a whole sentence, and 4) Many chunkers produce only one output, not all possible outputs in some ranked order.

2 UCSG Shallow Parsing Architecture

UCSG shallow parsing architecture is set within the UCSG full parsing framework for parsing natural language sentences which was initiated in the early 1990's at University of Hyderabad by Kavi Narayana Murthy (Murthy, 1995). In this paper, the focus is only on chunking - identifying chunks or word groups, handling ambiguities, and producing parses (chunk sequences) for given sentences. This can be extended to include thematic role assignment and clause structure analysis leading towards a full parser. Figure 1 shows the basic UCSG Shallow Parsing Architecture (Kumar and Murthy, 2006).

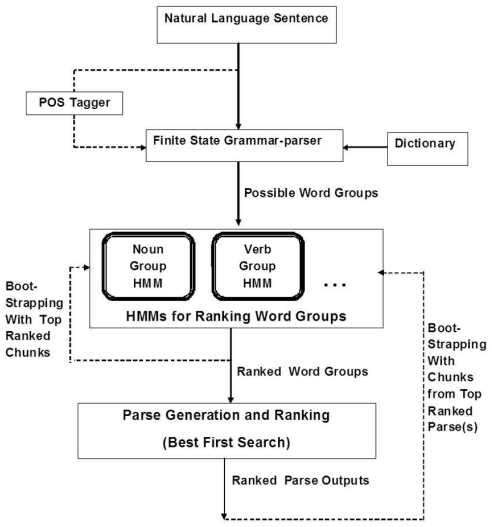


Figure 1: UCSG Shallow Parsing Architecture

The input to the parsing system is one sentence, either plain or POS tagged. Output is an ordered set of parses. Here by parse we mean a sequence of chunks that covers the given sentence with no overlaps or gaps. The aim is to produce all possible parses in ranked order hoping to get the best parse to the top.

A chunk or a "word group" as we prefer to call it in UCSG, is "a structural unit, a non-overlapping and non-recursive sequence of words, that can as a whole, play a role in some predication" (Murthy, 1995). Note that word groups do not include clauses (relative clauses, for example) or whole sentences. Every word group has a head which defines the type of the group. These word groups thus seem to be similar to *chunks* as generally understood (Molina and Pla, 2002; Sang and Buchholz, 2000; Megyesi, 2002). However, chunks in UCSG are required to correspond to thematic roles, which means for example, that prepositional phrases are handled properly. Many chunkers do not even build prepositional phrases - prepositions are treated as individual chunks in their own right. Thematic roles can be viewed from question-answering perspective. For example, in the sentence *'I teach at University of Hyderabad'*, *'at University of Hyderabad'* answers the 'where' question and should therefore be treated as a single chunk. It is well known that prepositional phrase attachment is a hard problem and the task we have set for ourselves here is thus significantly more challenging. The parse outputs in UCSG would be more semantic and hence should be better suited for many NLP applications.

In UCSG, a Finite State Grammar-Parser system generates all possible chunks in linear time. Chunk level HMMs are then used to rate and rank the chunks so produced. Finally, a kind of best first search strategy is applied to obtain chunk sequences hopefully in best first order. The aim is to develop wide coverage, robust parsing systems without need for a large scale parsed corpus to start with. Only a large POS tagged corpus is needed and a parsed corpus can be generated from within the architecture with minimal manual effort. Such a parsed corpus can be used for evaluation as also for further performance improvements.

We will need a dictionary which includes the frequency of occurrence of each word in each possible tag. Such a dictionary can be developed using a large POS tagged corpus.

2.1 Finite State Grammar-Parser

Here the task is only to *recognize* chunks and not produce a detailed description of the internal structure of chunks. Also, chunks by definition are non-recursive in nature, only linear order, repetition and optional items need to be considered. Finite state grammars efficiently capture linear precedence, repetition and optional occurrence of words in word groups. Finite state machines are thus both necessary and sufficient for recognizing word groups (Murthy, 1995). It is also well known that Finite State Machines are computationally efficient - linear time algorithms exist for recognizing word groups. All possible word groups can be obtained in a single left-to-right scan of the given sentence in linear time (Murthy, 1995). Finite state grammars are also conceptually simple and easy to develop and test.

The Finite State module accepts a sentence (either already POS tagged or tagged with all possible categories using the dictionary) and produces an unordered set of possible chunks taking into account all lexical ambiguities.

2.2 HMMs for Rating and Ranking Chunks

The second module is a set of Hidden Markov Models (HMMs) used for rating and ranking the word groups already produced by the Finite State Grammar-Parser. The hope is to get the best chunks near the top. This way, we are not pruning and yet we can hope to get the right chunks near the top and push down the others.

Words are observation symbols and POS tags are states in our HMMs. Formally, a HMM model $\lambda = (\pi, A, B)$ for a given chunk type can be described as follows:

Number of States (N) = number of relevant POS Categories

Number of Observation Symbols (M) = number of Words of relevant categories in the language

The initial state probability

$$\pi_i = P\{q_1 = i\} \tag{1}$$

where $1 \leq i \leq N$, q_1 is a category (state) starting a particular word group type.

State transition probability

$$a_{ij} = P\{q_{t+1} = j | q_t = i\} \tag{2}$$

where $1 \leq i, j \leq N$ and q_t denotes the category at time t and q_{t+1} denotes the category at time t+1.

Observation or emission probability

$$b_j(k) = P\{o_t = v_k | q_t = j\} \tag{3}$$

where $1 \leq j \leq N$, $1 \leq k \leq M$ and v_k denotes the k^{th} word, and q_t the current state.

We first pass a large POS tagged corpus through the Finite State module and obtain all possible chunks. Taking these chunks to be equi-probable, we estimate the HMM parameters by taking the ratios of frequency counts. One HMM is developed for each major category of chunks, say, one for noun-groups, one for verb-groups, and so on. The B matrix values are estimated from a dictionary that includes frequency counts for each word in every possible category. These initial models of HMMs are later refined using a bootstrapping technique as described later.

We simply estimate the probability of each chunk using the following equation :

$$P(O, Q|\lambda) = \pi_{q_1} b_{q_1}(o_1) a_{q_1,q_2} b_{q_2}(o_2) a_{q_2,q_3} \cdots a_{q_{t-1},q_t} b_{q_t}(o_t)$$

where q_1 ,q_2, $\cdots$, q_t is a state sequence, o_1 , o_2,$\cdots$, o_t is an observation sequence. Note that no Viterbi search involved here and the state sequence is also known. Thus even Forward/Backward algorithm is not required and rating the chunks is therefore computationally efficient.

The aim here is to assign the highest rank for the correct chunk and to push down other chunks. Since a final parse is a sequence of chunks that covers the given sentence with no overlaps or gaps, we evaluate the alternatives at each position in the sentence in a left-to-right manner.

Here, we use *Mean Rank Score* to evaluate the performance of the HMMs. Mean Rank Score is the mean of the distribution of ranks of correct chunks produced for a given training corpus. Ideally, all correct chunks would be at the top and hence the score would be 1. The aim is to get a Mean Rank Score as close to 1 as possible.

2.3 Parse Generation and Ranking

Parsing is a computationally complex task and generating all possible parses may be practically difficult. That is why, a generate-and-test approach

where we first generate all possible parses and then look for the correct parse among the parses produced is impracticable. Simply producing all or some parses in some random or arbitrary order is also not of much practical use. Many chunkers produce a single output which may or may not be correct. Here we instead propose a best first strategy wherein the very production of possible parses is in best first order and so, hopefully, we will get the correct parse within the top few and in practice we need not actually generate all possible parses at all. This way, we overcome the problems of computational complexity and at the same time avoid the risk of missing the correct parse if pruning is resorted to. Performance can be measured not only in terms of percentage of input sentences for which a fully correct parse is produced but also in terms of the rank of the correct parse in the top k parses produced, for any chosen value of k.

It may be noted that although we have already rated and ranked the chunks, simply choosing the locally best chunks at each position in a given sentence does not necessarily give us the best parse (chunk sequence) in all cases. Hence, we have mapped our parse selection problem into a graph search problem and used best first search algorithm to get the best parse for a given sentence.

Words and chunks in a sentence are referred to in terms of the positions they occupy in the sentence. Positions are marked between words, starting from zero to the left of the first word. The positions in the sentence are treated as nodes of the resulting graph. If a sentence contains N words then the graph contains $N + 1$ nodes corresponding to the $N + 1$ positions in the sentence. Word group $W_{i,j}$ is represented as an edge form node i to node j. We thus have a lattice structure. The cost of a given edge is estimated from the probabilities given by the HMMs. If and where a parsed training corpus is available, we can also use the transition probability from previous word group type to current word group type. It is possible to use the system itself to parse sentences and from that produce a manually checked parsed corpus with minimal human effort. We always start from the initial node 0. N is the goal node. Now our parse selection problem for a sentence containing N words becomes the task of finding an optimal (lowest cost) path from node 0 to node N.

We use the standard best first search algorithm. In best first search, we can inspect all the currently-available nodes, rank them on the basis of our partial knowledge and select the most promising of the nodes. We then expand the chosen node to generate it successors. The worst case complexity of best first search algorithm is exponential: $O(b^m)$, where b is the branching factor (i.e., the average number of nodes added to the open list at each level), and m is the maximum length of any path in the search space. As an example, a 40 word sentence has been shown to produce more than 10^{15} different parses (Kumar, 2007). In practice, however, we are usually interested in only the top k parses for some k and exhaustive search is not called for.

2.4 Bootstrapping

The HMM parameters can be refined through bootstrapping. We work with large data sets running into many hundreds of thousands of sentences and Baum-Welch parameter re-estimation would not be very practical. Instead, we use parsed outputs to re-build HMMs. By parsing a given sentence using the system and taking the top few parses only as training data, we can re-build HMMs that will hopefully be better. We can also simply use the top-ranked chunks for re-building the HMMs. This would reduce the proportion of invalid chunks in the training data and hence hopefully result in better HMM parameters. As can be seen from the results in the next section, this idea actually works and we can significantly improve the HMM parameters and improve parser performance as well.

3 Experiments and Results

The entire parsing system has been implemented in Perl under Linux. Extensive experimentation has been carried out to evaluate the performance of the system. However, direct comparisons with other chunkers and parsers are not feasible as the architectures are quite different. All the experiments have been carried out on a system with Pentium Core 2 DUO 1.86 GHz Processor and 1 GB RAM. Transcripts from the implemented system have been included in the next section.

3.1 Dictionary

We have developed a dictionary of 138,000 words including frequency of occurrence for each tag for each word. The dictionary includes derived words but not inflected forms. The dictionary has been built from the British National Corpus(BNC) (Burnard, 2000), an English text corpus of about 100 Million words. Closed class words have been manually checked. The dictionary has a coverage of 98% on the BNC corpus itself, 86% on the Reuters News Corpus (Rose et

al., 2002) (about 180 Million words in size), 96.36% on the Susanne parsed corpus (Sampson, 1995) and 95.27% on the Link parser dictionary.

3.2 Sentence Boundary Detection

We have developed a sentence segmentation module using the BNC corpus as training data. We have used *delimiter, prefix, suffix and after-word* as features and extracted patterns from the BNC corpus. Decision Tree algorithms have been used and an average F-Measure of 98.70% has been obtained, comparable to other published results. See (Htay et al., 2006) for more details.

3.3 Tag Set

We have studied various tag sets including BNC C5, BNC C7, Susanne and Penn Tree Bank tag sets. Since our work is based on BNC 1996 edition with C5 tag set, we have used C5 tag set and made some extensions as required. We now have a total of 71 tags in our extended tag set (Kumar, 2007).

3.4 Manually Parsed Corpus

We have developed a manually checked parsed corpus of 4000 sentences, covering a wide variety of sentence structures. Of these, 1000 sentences have been randomly selected from the BNC corpus, 1065 sentences from 'Guide to Patterns and Usage in English' (Hornby, 1975) and 1935 sentences from the CoNLL-2000 test data. This corpus is thus very useful for evaluating the various modules of the parsing architecture and also for bootstrapping.

This corpus was developed by parsing the sentences using this UCSG shallow parser itself and then manually checking the top parse and making corrections where required. Our experience shows that this way we can build manually checked parsed corpora with minimal human effort.

3.5 Tagging

If a POS tagger is available, we can POS tag the input sentences before sending them to the parser. Otherwise, all possible tags from the dictionary may be considered. In our work here, we have not used any POS tagger. All possible tags are assigned from our dictionary and a few major rules of inflectional morphology of English, including plurals for nouns, past tense, gerundial and participial forms of verbs and degrees of comparison for adjectives are handled. Unresolved words are assigned NP0 (Proper Name) tag.

3.6 Finite State Grammar

We have developed a Finite State Grammar for identifying English word groups. The Finite State Machine has a total of 50 states of which 24 are final states. See (Kumar, 2007) for further details.

The UCSG Finite State Grammar recognizes verb-groups, noun-groups, adverbial-groups, adjective-groups, to-infinitives, coordinate and subordinate conjunctions. There are no separate prepositional phrases - prepositions are treated as surface case markers in UCSG - their primary role is to indicate the relationships between chunks and the thematic roles taken up by various noun groups. Prepositional groups are therefore treated on par with noun groups.

We have evaluated the performance of the FSM module on various corpora - Susanne Parsed Corpus, CoNLL 2000 test data set and on our manually parsed corpus of 4000 sentences. The evaluation criteria is Recall (the percentage of correct chunks recognized) alone since the aim here is only to include the correct chunks. We have achieved a high recall of 99.5% on manually parsed corpus, 95.06% on CoNLL test data and 88.02% on Susanne corpus.

The reason for the relatively low Recall on the Susanne corpus is because of the variations in the definition of phrases in Susanne corpus. For example, Susanne corpus includes relative clauses into noun groups. The reasons for failures on CoNLL test data have been traced mainly to missing dictionary entries and inability of the current system to handle multi-token adverbs.

3.7 Building and Refining HMMs

HMMs were initially developed from 3.7 Million POS-tagged sentences taken from the BNC corpus. Sentences with more than 40 words were excluded. Since we use an extended C5 tag set, POS tags had to be mapped to the extended set where necessary. HMM parameters were estimated from the chunks produced by the Finite State grammar, taking all chunks to be equi-probable. Separate HMMs were built for noun groups, verb groups, adjective groups, adverb groups, infinitive groups and one HMM for all other chunk types.

The chunks produced by the FSM are ranked using these HMMs. It is interesting to observe the Recall and Mean Rank Score within the top k ranks, where k is a given cutoff rank. Table 1 shows that there is a clear tendency for the correct chunks to bubble up

close to the top. For example, more than 95% of the correct chunks were found within the top 5 ranks.

Table 1: Performance of the HMM Module on the Manually Parsed Corpus of 4000 sentences

Cut-off	Plain		POS Tagged	
	Mean Rank	Cumulative Recall (%)	Mean Rank	Cumulative Recall (%)
1	1	43.06	1	62.74
2	1.38	69.50	1.28	86.97
3	1.67	84.72	1.43	95.64
4	1.85	91.69	1.50	98.31
5	1.96	95.13	1.54	99.25

We have also carried out some experiments to see the effect of the size of training data used to build HMMs. We have found that as we use more and more training data, the HMM performance improves significantly, clearly showing the need for working with very large data sets. See (Kumar, 2007) for more details.

3.7.1 Bootstrapping

To prove the bootstrapping hypothesis, we have carried out several experiments. Plain text sentences from BNC corpus, 5 to 20 words in length, have been used. All possible chunks are obtained using the Finite State Grammar-Parser and HMMs built from these chunks. In one experiment, only the chunks rated highest by these very HMMs are taken as training data for bootstrapping. In a second experiment, best first search is also carried out and chunks from the top ranked parse alone are taken for bootstrapping. In a third experiment, data from these two sources have been combined. Best results were obtained when the chunks from the top parse alone were used for bootstrapping. Table 2 shows the effect of bootstrapping on the HMM module for plain sentences.

Table 2: Effect of Bootstrapping: on 4000 sentences from Manually Parsed Corpus containing a total of 27703 chunks

Cutoff	Iteration-1		Iteration-2	
	Recall	Mean Rank	Recall	Mean Rank
1	45.52	1.0	47.25	1.0
2	71.43	1.36	72.81	1.35
3	85.22	1.63	85.95	1.60
4	91.75	1.80	92.20	1.77
5	94.94	1.90	95.30	1.87

It may be observed that both the Recall and Mean Rank Scores have improved. Our experiments show that there is also some improvement in the final parse when the HMMs obtained through bootstrapping are used. These observations, seen consistently for both plain and POS tagged sentences, show the effectiveness of the overall idea.

3.8 Parse Generation and Ranking

It may be noted that in principle the performance of the parser in terms of its ability to produce the correct parse is limited only by the Finite State Grammar and the dictionary, since the other modules in the UCSG architecture do not resort to any pruning. However, in practical usage we generally impose a time limit or a cutoff and attempt to produce only the top k parses. In this latter case, the percentage of cases where the fully correct parse is included would be a relevant performance indicator. Percentage of correct chunks in the top parse is another useful indicator.

When tested on untagged sentences, on the 1065 linguistically rich sentence corpus forming part of the manually checked parsed corpus developed by us, the parser could generate fully correct parse within the top 5 parses in 930 cases, that is, 87.32% of the cases. In 683 cases the correct parse was the top parse, 146 correct parses were found in position 2, 56 in position 3, 29 in position 4 and 16 in position 5. Thus the mean rank of the correct parses is 1.44. There is a clear tendency for the correct parses to appear close to the top, thereby verifying the best first strategy. If top 10 parses are generated, correct parse is obtained in 52 more cases and the Mean Rank Score goes to 1.75.

We give below the performance on the whole of our 4000 strong manually checked corpus. Plain sentences and POS tagged sentences have been tested separately. The results are summarized in table 3. Here, we have restricted the parsing time taken by the best first search algorithm to 3 epoch seconds for each sentence.

Table 3: Performance of the Best First Search Module - Test Data of 4000 Sentences

Rank	No. of Correct Parses	
	(Plain)	(POS tagged)
1	1210	2193
2	352	495
3	157	164
4	83	129
5	68	91
% of Correct Parses in Top 5	46.75	76.80
% of Correct Chunks in Best Parse	83.92	88.26

In about 77% of the cases, the fully correct parse is found within the top 5 parses when the input sentences are POS tagged. Given the nature of chunks produced in UCSG, this is quite encouraging. In fact the top parse is nearly correct in many cases. Further experiments and manual evaluations are planned.

We have also observed that 96.01% of the words are assigned correct POS tags in the top parse. We observe that most of the times the top parse given by the parse generation module is almost correct.

Chunkers are usually evaluated just for the percentage of correct chunks they produce. We have placed greater demands on ourselves and we expect our parser to produce optimal chunk sequence for the whole sentence. Further, we produce all (or top few) combinations and that too in hopefully a best first order. Also, the very nature of chunks in UCSG makes the task more challenging. More over, we have used a fairly fine grained tag set with more than 70 tags. The data we have started with, namely the BNC POS tagged corpus, is far from perfect. Given these factors, the performance we are able to achieve both in terms of percentage of correct chunks in the top parse and rank of the fully correct parse is very encouraging.

4 Transcripts:

Here we give the actual transcripts from the system. For want of space, only a very simple example has been included. Stars have been added in the beginning of lines containing correct alternatives.

```
Input: I am studying at University of Hyderabad.

Tags from the Dictionary: <PNN_CRD><i>##<VBB><am>##<VVG>
    <studying>##<PRN_PRP_AVP><at>##<NN1><university>
    ##<PRN_PRF_AVP><of>##<NP0><Hyderabad>##

Chunks Recognized by the FSM:
<ng><0-1><CRD><i>
<ajg><0-1><CRD><i>
*<ng><0-1><PNN><i>
<vg><1-2><VBB><am>
*<vg><1-3><VBB><am>##<VVG><studying>
<vg><1-4><VBB><am>##<VVG><studying>##<AVP><at>
<vgs><2-3><VVG><studying>
<ng><2-3><VVG><studying>
<ajg><2-3><VVG><studying>
<vgs><2-4><VVG><studying>##<AVP><at>
<ng><2-5><VVG><studying>##<PRP><at>##<NN1><university>
<ng><2-7><VVG><studying>##<PRP><at>##<NN1><university>
    ##<PRF><of>##<NP0><hyderabad>
<part><3-4><AVP><at>
<ng><3-5><PRP><at>##<NN1><university>
*<ng><3-7><PRP><at>##<NN1><university>##<PRF><of>##
    <NP0><hyderabad>
<ng><4-5><NN1><university>
<ng><4-7><NN1><university>##<PRF><of>##<NP0><hyderabad>
<part><5-6><AVP><of>
<ng><5-7><PRF><of>##<NP0><hyderabad>
<ng><6-7><NP0><hyderabad>

Ranking by HMMs:
*<ng><0-1><PNN><i><-3.2491231040407><1><3><1>
<ng><0-1><CRD><i><-9.56376400947296><2><3><1>
<ajg><0-1><CRD><i><-36.8109739544272><3><3><1>
<vg><1-2><VBB><am><-7.27367328109116><1><3><2>
*<vg><1-3><VBB><am>##<VVG><studying><-15.945895214915>
    <2><3><2>
<vg><1-4><VBB><am>##<VVG><studying>##<AVP><at>
    <-25.5608664628101><3><3><2>
<vgs><2-3><VVG><studying><-10.5328994260119><1><6><3>
<ng><2-3><VVG><studying><-12.7929752284183><2><6><3>
<vgs><2-4><VVG><studying>##<AVP><at><-20.147870673907>
    <3><6><3>
<ng><2-5><VVG><studying>##<PRP><at>##<NN1><university>
    <-30.3473074722636><4><6><3>
<ajg><2-3><VVG><studying><-32.767076078699><5><6><3>
<ng><2-7><VVG><studying>##<PRP><at>##<NN1><university>##
    <PRF><of>##<NP0><hyderabad><-35.1643970692879><6><6><3>
<part><3-4><AVP><at><-7.99897865005313><1><3><4>
<ng><3-5><PRP><at>##<NN1><university><-15.7772256956695>
    <2><3><4>
*<ng><3-7><PRP><at>##<NN1><university>##<PRF><of>##<NP0>
    <hyderabad><-20.5943152926938><3><3><4>
<ng><4-5><NN1><university><-13.2259579687766><1><2><5>
<ng><4-7><NN1><university>##<PRF><of>##<NP0><hyderabad>
    <-18.0430475658009><2><2><5>
<part><5-6><AVP><of><-3.87313237166961><1><2><6>
<ng><5-7><PRF><of>##<NP0><hyderabad><-19.0843146188301>
    <2><2><6>
<ng><6-7><NP0><hyderabad><-3.43828759462479><1><1><7>

Final Parse:

*<ng>[<PNN><i>]</ng> <vg>[<VBB><am>##<VVG><studying>]</vg>
    <ng>[<PRP><at>##<NN1><university>##<PRF><of>##<NP0>
    <hyderabad>]</ng> -- -41.2629507152745

<ng>[<PNN><i>]</ng> <vg>[<VBB><am>]</vg> <ng>[<VVG>
    <studying>]</ng><ng>[<PRP><at>##<NN1><university>##<PRF>
    <of>##<NP0><hyderabad>]</ng> -- -46.7375549370651

<ng>[<PNN><i>]</ng> <vg>[<VBB><am>]</vg> <ng>[<VVG>
    <studying>##<PRP><at>##<NN1><university>##<PRF><of>##
    <NP0><hyderabad>]</ng> -- -47.1608105580448

<ng>[<CRD><i>]</ng> <vg>[<VBB><am>##<VVG><studying>]</vg>
    <ng>[<PRP><at>##<NN1><university>##<PRF><of>##<NP0>
    <hyderabad>]</ng> -- -47.5775916207068

<ng>[<PNN><i>]</ng> <vg>[<VBB><am>##<VVG><studying>##
    <AVP><at>]</vg><ng>[<NN1><university>##<PRF><of>##
    <NP0><hyderabad>]</ng> -- -48.3266542362767
```

5 Conclusions:

A hybrid architecture for developing wide coverage shallow parsing systems, without need for a large scale parsed corpus to start with, has been proposed and its effectiveness demonstrated by developing a wide coverage shallow parser for English. The system has been built and tested on very large data sets, covering a wide variety of texts, giving us confidence that the system will perform well on new, unseen texts. The system is general and not domain specific, but we can adapt and fine tune for any specific domain to achieve better performance. We are confident that wide coverage and robust shallow parsing systems can be developed using the UCSG architecture for other languages of the world as well. We plan to continue our work on English parsing while we also start our work on Telugu.

References

Steven P. Abney. 1991. *Parsing by Chunks*. Kluwer, Principle-Based Parsing: Computation and Psycholinguistics edition.

Steven P. Abney. 1996. Partial Parsing via Finite-State Cascades. In *Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information*, pages 8–15, Prag.

L. Burnard. 2000. The Users' Reference Guide for the British National Corpus. Oxford University Computing Services, Oxford.

Xavier Carreras and Lluys Marquez. 2003. Phrase Recognition by Filtering and Ranking with Perceptrons. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP-2003*, pages 127–132, Borovets, Bulgaria.

Herve Dejean. 2002. Learning Rules and their Exceptions. In *Journal of Machine Learning Research, Volume 2*, pages 669–693.

G. Grefenstette. 1996. Light Parsing as Finite State Filtering. In *Workshop on Extended Finite State Models of Language*, Budapest, Hungary.

A. S. Hornby. 1975. *Guide to Patterns and Usage in English*. Oxford University Press.

Hla Hla Htay, G. Bharadwaja Kumar, and Kavi Narayana Murthy. 2006. Constructing English-Myanmar Parallel Corpora. In *Proceedings of Fourth International Conference on Computer Applications*, pages 231–238, Yangon, Myanmar.

G Bharadwaja Kumar and Kavi Narayana Murthy. 2006. UCSG Shallow Parser. *Proceedings of CICLING 2006, LNCS*, 3878:156–167.

G. Bharadwaja Kumar. 2007. *UCSG Shallow Parser: A Hybrid Architecture for a Wide Coverage Natural Language Parsing System*. Phd thesis, University of Hyderabad.

B Megyesi. 2002. Shallow Parsing with PoS Taggers and Linguistic Features. In *Journal of Machine Learning Research, Volume 2*, pages 639–668.

Antonio Molina and Ferran Pla. 2002. Shallow Parsing using Specialized HMMs. In *Journal of Machine Learning Research, Volume 2*, pages 595–613.

Kavi Narayana Murthy. 1995. *Universal Clause Structure Grammar*. Phd thesis, University of Hyderabad.

Miles Osborne. 2002. Shallow Parsing using Noisy and Non-Stationary Training Material. In *Journal of Machine Learning Research, Volume 2*, pages 695–719.

E. Roche. 1997. *Parsing with Finite State Transducers*. MIT Press, finite-State Language Processing edition.

T.G. Rose, M. Stevenson, and M. Whitehead. 2002. The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, Las Palmas de Gran Canaria.

Geoffrey Sampson. 1995. *English for the Computer*. Clarendon Press (The Scholarly Imprint of Oxford University Press).

E. F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132, Lisbon, Portugal.

Erik F. Tjong Kim Sang. 2002. Memory-Based Shallow Parsing. In *Journal of Machine Learning Research, Volume 2*, pages 559–594.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. In Claire Cardie, Walter Daelemans, Claire Nedellec, and Erik Tjong Kim Sang, editors, *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132. Lisbon, Portugal.

Memory-Inductive Categorial Grammar: An Approach to Gap Resolution in Analytic-Language Translation

Prachya Boonkwan **Thepchai Supnithi**

Human Language Technology Laboratory
National Electronics and Computer Technology Center (NECTEC)
112 Thailand Science Park, Phaholyothin Road,
Khlong 1, Pathumthani 12120, Thailand
{prachya.boonkwan, thepchai.supnithi}@nectec.or.th

Abstract

This paper presents a generalized framework of syntax-based gap resolution in analytic language translation using an extended version of categorial grammar. Translating analytic languages into Indo-European languages suffers the issues of gapping, because "deletion under coordination" and "verb serialization" are necessary to be resolved beforehand. Rudimentary operations, i.e. antecedent memorization, gap induction, and gap resolution, were introduced to the categorial grammar to resolve gapping issues syntactically. Hereby, pronominal references can be generated for deletion under coordination, while sentence structures can be properly selected for verb serialization.

1 Background

Analytic language, such as Chinese, Thai, and Vietnamese, is any language whose syntax and meaning relies on particles and word orders rather than inflection. Pronouns and other grammatical information, such as tense, aspect, and number, expressed by use of adverbs and adjectives, are often omitted. In addition to *deletion under coordination* and *verb serialization*, called *gapping* (Hendriks, 1995), translation from analytic languages into Indo-European ones becomes a hard task because (1) an ordinary parser cannot parse some problematic gapping patterns and (2) these omissions are necessary to be resolved beforehand. We classify resolution of the issue into two levels: syntactic/semantic and pragmatic. Gapping, which we considered as a set of bound variables, can be resolved in syntactic/semantic level (Partee, 1975). Omission of other grammatical information is, on the contrary, to be resolved in pragmatic level because some extra-linguistic knowledge is required. Consequently, we concentrate in this paper the resolution of gapping by means of syntax and semantics.

Many proposals to gap resolution were introduced, but we classify them into two groups: non-ellipsis-based and ellipsis-based. *Non-ellipsis-based approach* is characterized by: (a) strong proof system (Lambek, 1958), and (b) functional composition and type raising that allow coordination of incomplete constituents, such as CG (Ajdukiewicz, 1935; Bar-Hillel, 1953; Moortgat, 2002), CCG (Steedman, 2000), and multimodal CCG (Baldridge and Kruijff, 2003). Proposals in this approach, such as (Hendriks, 1995; Jäger, 1998a; Jäger, 1998b), introduced specialized operators to resolve overt anaphora, while covert anaphora is left unsolved. *Ellipsis-based approach* is characterized by treating incomplete constituents as if they are of the same simple type but contain ellipsis inside (Yatabe, 2002; Cryssmann, 2003; Beavers and Sag, 2004). However, Beavers and Sag (2004) evidenced that ellipsis-based analysis possibly reduces the acceptability of language, because the resolution is *per se* completely uncontrolled.

In this paper, we introduce an integration of the two approaches that incorporates strong proof system and ellipsis-based analysis. Antecedent memorization and gap induction are introduced to imitate ellipsis-based analysis. The directions of ellipsis are

also used to improve the acceptability of language.

The rest of the paper is structured as follows. Section 2 describes the formalization of our method. Section 3 evidences the coverage of the framework on coping with the gapping issues in analytic languages. Section 4 further discusses coverage and limitations of the framework comparing with CG and its descendants. Section 5 explains relevance of the proposed formalism to MT. Finally, Section 6 concludes the paper and lists up future work.

2 Memory-Inductive Categorial Grammar

Memory-Inductive Categorial Grammar, abbreviated MICG, is a version of pure categorial grammar extended by ellipsis-based analysis. On the contrary, it relies on antecedent memorization, gap induction, and gap resolution that outperform CCG's functional composition and type raising.

All grammatical expressions of MICG are, like CG, distinguished by a syntactic category identifying them as either a function from arguments of one type to result another (a.k.a. *function*), or an argument (a.k.a. *primitive category*). Let us exemplify the MICG by defining an example grammar G below.

$$\begin{aligned} \mathsf{John}, \mathsf{Mary}, \mathsf{sandwich}, \mathsf{noodle} &\vdash np \\ \mathsf{eats} &\vdash (np\backslash s)/np \\ \mathsf{and} &\vdash \& \end{aligned}$$

The lexicons John, Mary, sandwich, and noodle are assigned with a primitive category np. The lexicon eats is assigned with a function that forms a sentence s after taking np from the right side ($/np$) and then taking np from the left side ($np\backslash$). The lexicon and is assigned with a conjunction category (&). By means of syntactic categories assigned to each lexicon, the derivation for a simple sentence 'John eats noodle' is shown in (1).

(1)
$$\dfrac{\dfrac{\mathsf{John}}{\mathsf{John}\vdash np} \quad \dfrac{\dfrac{\mathsf{eats}}{\mathsf{eats}\vdash (np\backslash s)/np} \quad \dfrac{\mathsf{noodle}}{\mathsf{noodle}\vdash np}}{\mathsf{eats}\circ\mathsf{noodle}\vdash np\backslash s}}{\mathsf{John}\circ(\mathsf{eats}\circ\mathsf{noodle})\vdash s}$$

CG suffers some patterns of coordination e.g. SVO&SO as exemplified in (2).

(2) John eats noodle, and Mary, sandwich.

One should find that the second conjunct cannot be reduced into s by means of CG, because it lacks of the main verb 'eats.' The main verb in the first conjunct should be remembered and then filled up to the ellipsis of the second conjunct to accomplish the derivation. This matter of fact motivated us to develop MICG by introducing to CG the process of remembering an antecedent from a conjunct, called *memorization*, and filling up an ellipsis in the other conjunct, called *induction*. There are three mandatory operations in MICG: antecedent memorization, gap induction, and gap resolution.

One of two immediate formulae combined in the derivation can be memorized as an antecedent. The resulted syntactic category is modalized by the modality $\square^D_F$, where D is a direction of memorization ($<$ for the left side and $>$ for the right side), and F is the memorized formula. The syntactic structure of the memorized formula is also modalized with the notation $\square$ to denote the memorization. It is restricted in MICG that the memorized formula must be unmodalized to maintain mild context-sensitivity. For example, let us consider the derivation of the first conjunct of (2), 'John eats noodle,' with antecedent memorization at the verb 'eats' in (3). As seen, a modalized formula can combine with another unmodalized formula while all modalities are preserved.

(3)
$$\dfrac{\dfrac{\mathsf{John}}{\mathsf{John}\vdash np} \quad \dfrac{\dfrac{\mathsf{eats}}{\square\mathsf{eats}\vdash (np\backslash s)/np} \quad \dfrac{\mathsf{noodle}}{\mathsf{noodle}\vdash np}}{\square\mathsf{eats}\circ\mathsf{noodle}\vdash \square^{<}_{\mathsf{eats}\vdash (np\backslash s)/np}(np\backslash s)}}{\mathsf{John}\circ(\square\mathsf{eats}\circ\mathsf{noodle})\vdash \square^{<}_{\mathsf{eats}\vdash (np\backslash s)/np}s}$$

Any given formula can be induced for a missing formula, or a *gap*, at any direction, and the induced gap contains a syntactic category that can be combined to that of the formula. The resulted syntactic category of combining the formula and the gap is modalized by the modality $\diamond^D_F$, where D is a direction of induction, and F is the induced formula at the gap. The syntactic structure of F is an uninstantiated variable and also modalized with the notation $\diamond$ to denote the induction. The induced formula is necessary to be unmodalized for mild context-sensitivity. For example, let us consider the derivation of the second conjunct of (2), 'Mary, sandwich,' with gap induction before the word 'sandwich' in (4). The vari-

able of syntactic structure will be resolved with an appropriate antecedent containing the same syntactic category in the gap resolution process.

(4)
$$\dfrac{\dfrac{\mathsf{Mary}}{\mathsf{Mary} \vdash np} \quad \dfrac{\mathsf{sandwich}}{\mathsf{sandwich} \vdash np}}{\dfrac{}{\Diamond X \circ \mathsf{sandwich} \vdash \Diamond^{<}_{X \vdash (np\backslash s)/np}(np\backslash s)}}}{\mathsf{Mary} \circ (\Diamond X \circ \mathsf{sandwich}) \vdash \Diamond^{<}_{X \vdash (np\backslash s)/np} s}$$

Gap resolution matches between memorized antecedents and induced gaps to associate ellipses to their antecedents during derivation of coordination and serialization. That is, two syntactic categories $\Box^{D_1}_{F_1}C$ and $\Diamond^{D_2}_{F_2}C$ are matched up and canceled from the resulted syntactic category, if they have the same syntactic categories C, their directions D_1 and D_2 are equal, and their memorized/induced formulae F_1 and F_2 are unified. For example, let us consider the derivation of 'John eats noodle, and Mary, sandwich' in Figure 1. The modalities $\Box^{<}_{\mathsf{eats} \vdash (np\backslash s)/np} s$ and $\Diamond^{<}_{X \vdash (np\backslash s)/np} s$ are matched up together. Their memorized/induced formulae are also unified by instantiating the variable X with 'eats'. Eventually, after combining them and the conjunction 'and,' the derivation yields out the formula $(\mathsf{John} \circ (\Box \mathsf{eats} \circ \mathsf{noodle})) \circ (\mathsf{and} \circ (\mathsf{Mary} \circ (\Diamond \mathsf{eats} \circ \mathsf{sandwich}))) \vdash s$.

Gap resolution could also indicate argument sharing in coordination and serialization. $\Diamond^{D_1}_{F_1}C$ and $\Diamond^{D_2}_{F_2}C$ can be also matched up, if they have the same syntactic categories C, their directions D_1 and D_2 are equal, and their memorized/induced formulae F_1 and F_2 are unified. However, they must be preserved in the resulted syntactic category. For example, let us consider the derivation in Figure 2. By means of unification of induced formulae, the variables X and Y are unified into the variable Z.

A formal definition of MICG is given in Appendix A. MICG is applied to resolve deletion under coordination and serialization in analytic languages in the next section.

3 Gap Resolution in Analytic Languages

There are two causes of gapping in analytic languages: coordination and serial verb construction. Each of which complicates the analysis module of MT to resolve such issue before transferring. In this section, problematic gapping patterns are analyzed in forms of generalized patterns by MICG. For simplification reason, syntactic structure is suppressed during derivation.

3.1 To resolve gapping under coordination

Coordination in analytic languages is more complex than that of Indo-European ones. Multi-conjunct coordination is suppressed here because biconjunct coordination can be applied. Besides SVO&VO and SV&SVO patterns already resolved by CCG (Steedman, 2000), there are also SVO&SV, SVO&V, SVO&SO (already illustrated in Figure 1), and SVO&SA patterns.

The pattern SVO&SV exhibits ellipsis at the object position of the second conjunct. The analysis of SVO&SV is illustrated in (5). It shows that the object of the first conjunct is memorized while the verb of the second conjunct is induced for the object.

(5)
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{S}}{np} \quad \dfrac{\dfrac{\mathrm{V}}{(np\backslash s)/np} \quad \dfrac{\mathrm{O}}{np}}{\Box^{>}_{np}(np\backslash s)}}{\Box^{>}_{np} s}}{} \quad \dfrac{\dfrac{\mathrm{\&}}{\mathrm{\&}} \quad \dfrac{\dfrac{\mathrm{S}}{np} \quad \dfrac{\mathrm{V}}{(np\backslash s)/np}}{\Diamond^{>}_{np}(np\backslash s)}}{\Diamond^{>}_{np} s}}{s}$$

Analysis of the sentence pattern SVO&V, illustrated in (6), exhibits ellipses at the subject and the object positions of the second conjunct. The subject and the object of the first conjunct are memorized, while the verb of the second conjunct is induced twice for the object and for the subject, respectively.

(6)
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{S}}{np} \quad \dfrac{\dfrac{\mathrm{V}}{(np\backslash s)/np} \quad \dfrac{\mathrm{O}}{np}}{\Box^{>}_{np}(np\backslash s)}}{\Box^{<}_{np}\Box^{>}_{np} s}}{} \quad \dfrac{\dfrac{\mathrm{\&}}{\mathrm{\&}} \quad \dfrac{\dfrac{\mathrm{V}}{(np\backslash s)/np}}{\Diamond^{>}_{np}(np\backslash s)}}{\Diamond^{<}_{np}\Diamond^{>}_{np} s}}{s}$$

The pattern SVO&SA exhibits ellipsis at the predicate position of the second conjunct, because only the adverb (A) is left. Suppose the adverb, typed $(np\backslash s)/(np\backslash s)$, precedes the predicate. Illustrated in (7), the predicate of the first conjunct is memorized, while the adverb of the second conjunct is inducted for the predicate.

(7)
$$\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{S}}{np} \quad \dfrac{\dfrac{\mathrm{V}}{(np\backslash s)/np} \quad \dfrac{\mathrm{O}}{np}}{np\backslash s}}{\Box^{>}_{np\backslash s} s}}{} \quad \dfrac{\dfrac{\mathrm{\&}}{\mathrm{\&}} \quad \dfrac{\dfrac{\mathrm{S}}{np} \quad \dfrac{\mathrm{A}}{(np\backslash s)/(np\backslash s)}}{\Diamond^{>}_{np\backslash s}(np\backslash s)}}{\Diamond^{>}_{np\backslash s} s}}{s}$$

$$\dfrac{\dfrac{\text{John eats noodle}}{\text{John}\circ(\Box\text{eats}\circ\text{noodle})\vdash\Box^{<}_{\text{eats}\vdash(np\backslash s)/np}s}\quad\dfrac{\text{and}}{\text{and}\vdash\&}\quad\dfrac{\text{Mary, sandwich}}{\text{Mary}\circ(\Diamond X\circ\text{sandwich})\vdash\Diamond^{<}_{X\vdash(np\backslash s)/np}s}}{(\text{John}\circ(\Box\text{eats}\circ\text{noodle}))\circ(\text{and}\circ(\text{Mary}\circ(\Diamond\text{eats}\circ\text{sandwich})))\vdash s}$$

Figure 1: Derivation of 'John eats noodle, and Mary, sandwich.'

$$\dfrac{\dfrac{\text{eats noodle}}{\Diamond X\circ(\text{eats}\circ\text{noodle})\vdash\Diamond^{<}_{X\vdash np}s}\quad\dfrac{\text{and}}{\text{and}\vdash\&}\quad\dfrac{\text{drinks coke}}{\Diamond Y\circ(\text{drinks}\circ\text{coke})\vdash\Diamond^{<}_{Y\vdash np}s}}{(\Diamond Z\circ(\text{eats}\circ\text{noodle}))\circ(\text{and}\circ(\Diamond Z\circ(\text{drinks}\circ\text{coke})))\vdash\Diamond^{<}_{Z\vdash np}s}$$

Figure 2: Preservation of modalities in derivation

3.2 To resolve gapping under serial verb construction

Serial verb construction (SVC) (Baker, 1989) is construction in which a sequence of verbs appears in what seems to be a single clause. Usually, the verbs have a single structural object and share logical arguments (Baker, 1989). Following (Li and Thompson, 1981; Wang, 2007; Thepkanjana, 2006), we classify SVC into three main types: consecutive/concurrent events, purpose, and circumstance.

No operation specialized for tracing antecedent projection in consecutive/concurrent event construction has been proposed in CG or its descendants. In MICG, the serialization operation is specialized for this construction. For example, a Chinese sentence from (Wang, 2007) in (8) is analyzed as in (9).

(8)
tā mǎi piào jīn qù
he buy ticket enter go
'He buys a ticket and then goes inside.'

(9)
$$\dfrac{\dfrac{\dfrac{\text{tā}}{np}\quad\dfrac{\text{mǎi}}{(np\backslash s)/np}\quad\dfrac{\text{piào}}{np}}{np\backslash s}}{\Box^{<}_{np}s}\quad\dfrac{\dfrac{\dfrac{\text{jīn}}{np\backslash s}\quad\dfrac{\text{qù}}{np\backslash s}}{\Diamond^{<}_{np}s\quad\Diamond^{<}_{np}s}}{\Diamond^{<}_{np}s}}{s}$$

Illustrated in (9), the subject argument tā 'he' is projected through the verb sequence by means of memorization and induction modalities.

Purpose construction can also be handled by MICG. For example, a Thai sentence in (10) is analyzed as in (11).

(10)
kʰăʊ tɔ̀ː tʰɔ̂ː paj ɕʰáj naj bâːn
he attach pipe go use in house
'He attaches pipes to use in the house.'

(11)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{kʰăʊ}}{np}\quad\dfrac{\dfrac{\text{tɔ̀ː}}{(np\backslash s)/np}\quad\dfrac{\text{tʰɔ̂ː}}{np}}{\Box^{>}_{np}(np\backslash s)}}{\Box^{<}_{np}\Box^{>}_{np}s}}{\Box^{<}_{np}\Box^{>}_{np}s}\quad\dfrac{\dfrac{\dfrac{\text{paj}}{s\backslash s}\quad\dfrac{\dfrac{\text{ɕʰáj}}{(np\backslash s)/np}\quad\dfrac{\text{naj}}{(s\backslash s)/np}}{s\backslash s}}{\Diamond^{>}_{np}(np\backslash s)}}{\Diamond^{<}_{np}\Diamond^{>}_{np}s}}{\Box^{<}_{np}\Box^{>}_{np}s}\quad\dfrac{\dfrac{\text{bâːn}}{np}}{\Diamond^{<}_{np}\Diamond^{>}_{np}s}}{s}$$

Illustrated in (11), the two logical arguments, i.e. the subject kʰăʊ 'he' and the object tʰɔ̂ː 'pipe,' are projected through the construction.

SVC expressing circumstance of action is syntactically considered much as consecutive event construction. For example, a Chinese sentence from (Wang, 2007) in (12) is analyzed as in (13).

(12)
wǒ yòng kuàizi chī fàn
I use chopstick eat meal
'I eat meal with chopsticks.'

(13)
$$\dfrac{\dfrac{\dfrac{\text{wǒ}}{np}\quad\dfrac{\dfrac{\text{yòng}}{(np\backslash s)/np}\quad\dfrac{\text{kuàizi}}{np}}{np\backslash s}}{\Box^{<}_{np}s}\quad\dfrac{\dfrac{\dfrac{\text{chī}}{(np\backslash s)/np}\quad\dfrac{\text{fàn}}{np}}{np\backslash s}}{\Diamond^{<}_{np}s}}{s}$$

4 Coverage and Limitations

Proven in Theorem 1 in Appendix A, memorized constituents and induced constituents are cross-serially associated. Controlled by order and direction, each memorized constituent is guaranteed to be cross-serially associated to its corresponding induced gap, while each gap pair is also cross-serially associated revealing argument sharing. This causes cross-serial association, illustrated in Figure 3, among memorized constituents and induced gaps. Since paired modalities are either eliminated or preserved and no modalities are left on the start

symbol, it guarantees that there is eventually no modality in derivation. In conclusion, no excessive gap is over-generated in the language.

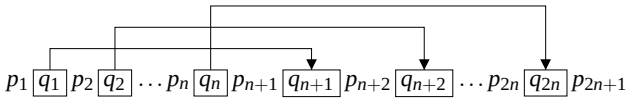


Figure 3: Cross-serial association

MICG's antecedent memorization and gap induction perform well in handling node raising. Node raising is analyzed in terms of MICG by memorizing the raised constituent at the conjunct it occurs and inducing a gap at the other conjunct. For example, the right node 'ice cream' is raised in the sentence 'I like but you don't like *ice cream*.' The sentence can be analyzed in terms of MICG in (14).

(14)

I	like	but	you	don't like	ice cream
np	$(np\backslash s)/np$	&	np	$(np\backslash s)/np$	np
	$\diamond^{>}_{np}(np\backslash s)$			$\square^{>}_{np}(np\backslash s)$	
	$\diamond^{>}_{np}s$			$\square^{>}_{np}s$	
			s		

Topicalization and contraposition are still the issues to be concerned for coverage over CCG. For example, in an example sentence 'Bagels, Yo said that Jan likes' from (Beavers and Sag, 2004), the NP 'Bagels' is topicalized from the object position of the relative clause's complement. (15) shows unparsability of the sentence.

(15)

Bagels,	Yo	said	that	Jan	likes
np	np	$(np\backslash s)/cl$	cl/s	np	$(np\backslash s)/np$
					$\diamond^{>}_{np}(np\backslash s)$
				$\diamond^{>}_{np}s$	
			$\diamond^{>}_{np}s$		
		$\diamond^{>}_{np}(np\backslash s)$			
	$\diamond^{>}_{np}s$				
		*****			

Furthermore, constituent shifting, such as dative shift and adjunct shift, is not supported by MICG. We found that it is also constituent extraction as consecutive constituents other than the shifted one are extracted from the sentence. For example, the adjunct 'skillfully' is shifted next to the main verb in the sentence 'Kahn blocked skillfully a powerful shot by Ronaldo' from (Baldridge, 2002) in (16).

(16)

Kahn	blocked	skillfully	a powerful shot by Ronaldo
np	$(np\backslash s)/np$	$(np\backslash s)\backslash(np\backslash s)$	np
	$\diamond^{>}_{np}(np\backslash s)$		
	$\diamond^{>}_{np}(np\backslash s)$		
	$\diamond^{>}_{np}s$		
	*****		

Since MICG was inspired by reasons other than those of CCG, the coverage of MICG is therefore different from CCG. Let us compare CG, CCG, and MICG in Table 1. CCG initially attempted to handle linguistic phenomena in English and other Indo-European languages, in which topicalization and dative shift play an important role. Applied to many other languages such as German, Dutch, Japanese, and Turkish, CCG is still unsuitable for analytic languages. MICG instead was inspired by deletion under coordination and serial verb construction in analytic languages. We are in progress to develop an extension of MICG that allows topicalization and dative shift avoiding combinatoric explosion.

5 Relevance to RBMT

Major issues of MT from analytic languages into Indo-European ones include three issues: anaphora generation, semantic duplication, and sentence structuring. Both syntax and semantics are used to solve such problems by MICG's capability of gap resolution. Case studies from our RBMT are exemplified for better understanding.

Our Thai-English MT system is rule-based and consists of three modules: analysis, transfer, and generation. MICG is used to tackle sentences with deletion under coordination and SVC which cannot be parsed by ordinary parsers. For good speed efficiency, an MICG parser was implemented in GLR-based approach and used to analyze the syntactic structure of a given sentence before transferring. The parser detects zero anaphora and resolves their antecedents in coordinate structure, and reveals argument sharing in SVC. Therefore, coordinate structure and SVC can be properly translated.

No experiment has been done on our system yet, but we hope to see an improvement of translation quality. We planned to evaluate the translation accuracy by using both statistical and human methods.

Table 1: Coverage comparison among CG, CCG, and MICG (Y = supported, N = not supported)

Linguistic phenomena	**CG**	**CCG**	**MICG**
Basic application	Y	Y	Y
Node raising	N	Y	Y
Topicalization/contraposition	N	**Y**	N
Constituent shifting	N	**Y**	N
Deletion under coordination	N	N	**Y**
Serial verb construction	N	N	**Y**

5.1 Translation of deletion under coordination

Coordinate structures in Thai drastically differ from those of English. This is because Thai allows zero anaphora at subject and object positions while English does not. Pronouns and VP ellipses must therefore be generated in place of deletion under coordination for grammaticality of English. Moreover, semantic duplication is often made use to emphasize the meaning of sentence, but its direct translation becomes redundant.

MICG helps us detect zero anaphora and resolve their antecedents, so that appropriate pronouns and ellipses can be generated at the right positions. By tracing resolved antecedents and ellipses, argument projections are disclosed and they can be used to control verb fusion. We exemplify three cases of translation of coordinate structure.

Case 1: Pronouns are generated to maintain grammaticality of English translation if the two verbs are not postulated in the verb-fusion table. For example, a Thai sentence in (17) is translated, while pronouns 'he' and 'it' are generated from Thai NPs nák·rian 'student' and kʰà·nŏm 'candy,' respectively.

(17)

nák·rian$_S$	sɯ́ː$_V$	kʰà·nŏm$_O$	lɛ́ːʊ$_{\&}$	kin$_V$
student	buy	candy	then	eat

'A student buys candy, then *he* eats *it*.'

Case 2: Two verbs V_1 and V_2 are fused togetherif they are postulated in the verb-fusion table to eliminate semantic duplication in English translation. The object form of S_2 is necessary to be generated in some cases. For example, in (18), the translation becomes 'He reports her this matter' instead of 'He tells her to know this matter.' Two verbs bɔ̀ːk 'tell' and sâːb 'know' are fused into a single verb 'report.' The object form of 'she,' '*her*,' is also generated.

(18)

kʰăʊ$_S$	bɔ̀ːk$_V$	hâj$_{\&}$	thəː$_S$	sâːp$_V$	rûːəŋ níː$_O$
he	tell	TO	she	know	this matter

'He *reports her* this matter.'

Case 3: A VP ellipsis is generated to maintain English grammaticality. For example, in (19), a VP ellipsis '*do*' is generated from a Thai VP mâi ɕʰɔ̂ːb don·triː rɔ́k 'not like rock music.'

(19)

ɕɔːn$_S$	ɕʰɔ̂ːp$_V$	don·triː rɔ́k$_O$	tɛ̀ː$_{\&}$	ɕʰăn$_S$	mâi$_A$
John	like	rock music	but	I	not

'John likes rock music, but I *do* not.'

5.2 Translation of SVC

Sentence structuring is also nontrivial for translation of Thai SVC. Thai uses SVC to describe consecutive/concurrent events, purposes, and circumstances. On the other hand, English describes each of those with different sentence structure. A series of verbs with duplicated semantics can be also clustered to emphasize the meaning of sentence in Thai, while English does not allow this phenomenon.

Because MICG reveals argument sharing in SVC, appropriate sentence structures can be selected by tracing argument sharing between two consecutive verbs. We exemplify two cases of translation of SVC.

Case 1: The second verb is participialized if the first verb is intransitive and its semantic concept is an action. For example, the present participial form of the verb 'see,' '*seeing*,' is generated in (20) .

(20)

sŏm·ɕʰaːj$_S$	dəːn$_V$	ɕʰom$_V$	pʰâːp·kʰiăn$_O$
Somchai	walk	see	paintings

'Somchai walks *seeing* paintings.'

Case 2: If the two cases above do not apply to the two verbs, they are translated directly by default. The conjunction 'and' is automatically added

to conjoin two verb phrases. In case of multiple-conjunct coordination, the conjunction will be added only before the last conjunct. For example, in (21), a pronoun '*it*' is generated from the NP kʰóːk 'coke,' while the conjunction 'and' is automatically added.

(21)

$pʰîː˙sǎː\upsilon_S$	$súː_V$	$kʰóːk_O$	$dùːm_V$
my elder sister	buy	coke	drink

'My elder sister buys coke *and* drinks *it*.'

6 Conclusion and Future Work

This paper presents Memory-Inductive Categorial Grammar (MICG), an extended version of categorial grammar, for gap resolution in analytic language translation. Antecedent memorization, gap induction, and gap resolution, are proposed to cope with deletion under coordination and serial verb construction. By means of MICG, anaphora can be generated for deletion under coordination, while sentence structure can be properly selected for serial verb construction. No experiment has been done to show improvement of translation quality by MICG.

The following future work remains. First, we will experiment on our Thai-English RBMT to measure improvement of translation quality. Second, criteria for pronominal reference generation in place of deletion under coordination will be studied. Third, once serial verb construction is analyzed, criteria of sentence structuring will further be studied based on an analysis of antecedent projection. Fourth and finally, constituent extraction and the use of extraction direction in the extraction resolution will be studied to avoid combinatoric explosion.

References

K. Ajdukiewicz. 1935. Die Syntaktische Konnexität. *Polish Logic*, pages 207–231.

M. C. Baker. 1989. Object Sharing and Projection in Serial Verb Constructions. *Linguistic Inquiry*, 20:513–553.

J. Baldridge and G. J. M. Kruijff. 2003. Multimodal combinatory categorial grammar. In *Proceedings of the 10th Conference of the European Chapter of the ACL 2003*, Budapest, Hungary.

J. Baldridge. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Y. Bar-Hillel. 1953. A Quasi-Arithmetical Notation for Syntactic Description. *Language*, 29:47–58.

J. Beavers and I. A. Sag. 2004. Coordinate ellipsis and apparent non-constituent coordination. In *Proceedings of the HPSG04 Conference*. Center for Computational Linguistics, Katholieke Universiteit Leuven, CSLI Publications.

B. Cryssmann. 2003. An asymmetric theory of peripheral sharing in HPSG: Conjunction reduction and coordination of unlikes. In *Proceedings of Formal Grammar Conference*.

P. Hendriks. 1995. Ellipsis and multimodal categorial type logic. In *Proceedings of Formal Grammar Conference*. Barcelona, Spain.

G. Jäger. 1998a. Anaphora and ellipsis in type-logical grammar. In *Proceedings of the 1th Amsterdam Colloquium*, Amsterdam, the Netherland. ILLC, Universiteit van Amsterdam.

G. Jäger. 1998b. Anaphora and quantification in categorial grammar. In *Lecture Notes in Computer Science; Selected papers from the 3rd International Conference, on logical aspects of Computational Linguistics*, volume 2014, pages 70–89.

J. Lambek. 1958. The Mathematics of Sentence Structure. *American Mathematical Monthly*, 65:154–170.

C. N. Li and S. A. Thompson. 1981. *Mandarin Chinese: A Functional Reference Grammar*. Berkeley: University of California Press.

M. Moortgat. 2002. Categorial grammar and formal semantics. In *Encyclopedia of Cognitive Science*, volume 1, pages 435–447. Nature Publishing Group.

B. H. Partee. 1975. Bound variables and other anaphors. In *Theoretical Issues in Natural Language Processing-2 (TINLAP-2)*, pages 79–85, University of Illinois at Urbana Champaign, July.

M. Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, Massachusetts.

K. Thepkanjana. 2006. Properties of events expressed by serial verb constructions in Thai. In *Proceedings of the 11th Biennial Symposium: Intertheoretical Approaches to Complex Verb Constructions*, Rice University.

X. Wang. 2007. Notes about Serial Verb Constructions in Chinese. *California Linguistic Notes*, 32(1).

S. Yatabe. 2002. A linearization-based theory of summative agreement in peripheral-node raising constructions. In *Proceedings of the HPSG02 Conference*, Standford, California. CSLI Publications.

A Formal Definition of MICG

Definition 1 (Closure of MICG) *Let V_A of category symbols, a finite set V_T of terminal symbols, and a set of directions $D = \{<,>\}$.*

The set C of all category symbols is given by: (1) For all $x \in V_A$, $x \in C$. (2) If $x,y \in C$, then so are $x\backslash y$ and x/y. (3) If $x \in C$, then so are $\Box^{<}_f x$, $\Box^{>}_f x$, $\Diamond^{<}_f x$, and $\Diamond^{>}_f x$, where $f \in F$ is a formula (described below). (4) Nothing else is in C.

The set T of all grammatical structures is given by: (1) For all $x \in V_T$, $x \in T$. (2) If $x,y \in T$, then so are $x \circ y$. (3) If $x \in T$, then so are $\Box x$ and $\Diamond x$. (4) Nothing else is in T.

The set F of all formulae is a set of terms $t \vdash x$, where $t \in T$ and $x \in C$. The set Q of all modalities is a set of all terms $\Box^{<}_f$, $\Box^{>}_f$, $\Diamond^{<}_f$, and $\Diamond^{>}_f$, where $f \in F$.

Definition 2 (Modality resolution) *For any directions $d \in D$, any formulae $f \in F$, and any modality sequences $\mathbf{M},\mathbf{M}_1,\mathbf{M}_2 \in Q^*$, the function $\oplus : Q^* \times Q^* \mapsto Q^*$ is defined as follows:*

$$\begin{aligned}
\Box^d_f\mathbf{M}_1 \oplus \Diamond^d_f\mathbf{M}_2 &\equiv \mathbf{M}_1 \oplus \mathbf{M}_2\\
\Diamond^d_f\mathbf{M}_1 \oplus \Box^d_f\mathbf{M}_2 &\equiv \mathbf{M}_1 \oplus \mathbf{M}_2\\
\Box^d_f\mathbf{M}_1 \oplus \Box^d_f\mathbf{M}_2 &\equiv \Box^d_f(\mathbf{M}_1 \oplus \mathbf{M}_2)\\
\Diamond^d_f\mathbf{M}_1 \oplus \Diamond^d_f\mathbf{M}_2 &\equiv \Diamond^d_f(\mathbf{M}_1 \oplus \mathbf{M}_2)\\
\varepsilon \oplus \mathbf{M} \equiv \mathbf{M} \oplus \varepsilon &\equiv \mathbf{M}
\end{aligned}$$

Definition 3 (MICG) *A memory-inductive categorial grammar (MICG) is defined as a quadruple $G = \langle V_T, V_A, s, R\rangle$, where: (1) V_T and V_A are as above. (2) $s \in V_A$ is the designated symbol called 'start symbol.' (3) $R : V_T \mapsto P(F)$ is a function assigning to each terminal symbol a set of formulae from F. The set of all strings generated from G is denoted as $L(G)$.*

Definition 4 (Acceptance of strings) *For any formulae $x,y \in F$, any grammatical structures $t_1,t_2,t_3 \in T$, any variables v of grammatical structures, and any modality sequences $\mathbf{M},\mathbf{M}_1,\mathbf{M}_2 \in Q^*$, the binary relation $\models \subseteq F^* \times F$ controls combination of formulae as follows:*

$$\begin{aligned}
t_1 \vdash y \quad t_2 \vdash y\backslash x &\models t_1 \circ t_2 \vdash x\\
t_1 \vdash x/y \quad t_2 \vdash y &\models t_1 \circ t_2 \vdash x\\
t_1 \vdash y \quad t_2 \vdash \mathbf{M}y\backslash x &\models \Box t_1 \circ t_2 \vdash \Box^{<}_{t_1\vdash y}\mathbf{M}x\\
t_1 \vdash \mathbf{M}y \quad t_2 \vdash y\backslash x &\models t_1 \circ \Box t_2 \vdash \Box^{>}_{t_2\vdash y\backslash x}\mathbf{M}x\\
t_1 \vdash x/y \quad t_2 \vdash \mathbf{M}y &\models \Box t_1 \circ t_2 \vdash \Box^{<}_{t_1\vdash x/y}\mathbf{M}x\\
t_1 \vdash \mathbf{M}x/y \quad t_2 \vdash y &\models t_1 \circ \Box t_2 \vdash \Box^{>}_{t_2\vdash y}\mathbf{M}x\\
t_2 \vdash \mathbf{M}y\backslash x &\models \Diamond v \circ t_2 \vdash \Diamond^{<}_{v\vdash y}\mathbf{M}x\\
t_1 \vdash \mathbf{M}y &\models t_1 \circ \Diamond v \vdash \Diamond^{>}_{v\vdash y\backslash x}\mathbf{M}x\\
t_2 \vdash \mathbf{M}y &\models \Diamond v \circ t_2 \vdash \Diamond^{<}_{v\vdash x/y}\mathbf{M}x\\
t_1 \vdash \mathbf{M}x/y &\models t_1 \circ \Diamond v \vdash \Diamond^{>}_{v\vdash y}\mathbf{M}x\\
t_1 \vdash \mathbf{M}_1x \quad t_3 \vdash \& \quad t_2 \vdash \mathbf{M}_2x &\models t_1 \circ (t_3 \circ t_2) \vdash (\mathbf{M}_1 \oplus \mathbf{M}_2)x\\
t_1 \vdash \mathbf{M}_1x \quad t_2 \vdash \mathbf{M}_2x &\models t_1 \circ t_2 \vdash (\mathbf{M}_1 \oplus \mathbf{M}_2)x
\end{aligned}$$

The binary relation $\Rightarrow \subseteq F^* \times F^*$ holds between two strings of formulae $\alpha X\beta$ and $\alpha Y\beta$, denoted $\alpha X\beta \Rightarrow \alpha Y\beta$, if and only if $X \models Y$, where $X,Y,\alpha,\beta \in F^*$ and $|X| \geq |Y|$. The relation $\Rightarrow^*$ is the reflexive transitive closure of $\Rightarrow$.

A string $w \in V_T^*$ is generated by G, denoted by $w \in L(G)$, if and only if $w = w_1 \ldots w_n$ and there is some sequence of formulae $f_1 \ldots f_n$ such that $f_i \in R(w_i)$ for all $1 \leq i \leq n$, and $f_1 \ldots f_n \Rightarrow^* s$. That is, $w_1 \ldots w_n$ is generated if and only if there is some choice of formula assignments by R to the symbols in $w_1 \ldots w_n$ that reduces to s.

Definition 5 *Correspondence between a grammatical structure and its syntactic category can be viewed as a tree with specialized node types. Each node is represented (m,S), where m is a node type $\{\emptyset,\Box,\Diamond\}$, and S is a modality sequence attached to the node's syntactic category.*

Definition 6 *A node that has the type m is said to be* marked *m where $m \in \{\Box, \Diamond\}$, while a node that has the type $\emptyset$ is said to be* unmarked.

Definition 7 *The function $\tau : Q \mapsto \{\Box,\Diamond\}$ maps a modality to a node modality, where $\tau(\Box^d_f) = \Box$ and $\tau(\Diamond^d_f) = \Diamond$ for all $d \in D$ and $f \in F$.*

Definition 8 *A substring generated from a node marked $\tau(\boldsymbol{M})$ beneath the node n is said to be* unpaired under *n, if and only if n has the modality sequence S and $\boldsymbol{M} \in S$.*

Definition 9 *Every string w generated from MICG can be rewritten in the form $w = p_1q_1 \ldots p_lq_lp_{l+1}q_{l+1} \ldots p_{2l}q_{2l}p_{2l+1}$, where q_i is a substring unpaired under n, p_j is a substring generated from unmarked nodes beneath n, $1 \leq i \leq l$, $1 \leq j \leq l+1$, and $l \geq 0$.*

Theorem 1 (Cross-serial association) *For every string generated from MICG $w = p_1q_1 \ldots p_lq_lp_{j(l)}q_{j(1)} \cdots p_{j(l)}q_{j(l)}p_{j(l)+1}$, every couple q_i and $q_{j(i)}$ are associated by $\oplus$ for all $1 \leq i \leq l$, where $j(i) = l+i$ and $l \geq 0$.*

Proof Let us prove this property by mathematical induction.

Basic step: Let $l = 0$. We obtain that $w_0 = p_1$. Since there is no unpaired substring, this case is trivially proven.

Hypothesis: Let $l = k$. Suppose that $w_k = p_1q_1 \ldots p_{j(k)}q_{j(k)}p_{j(k)+1}$. We rewrite $w_k = w^1_kw^2_k$, where $w^1_k = p_1q_1 \ldots p_kq_kp'_{j(1)}$ and $w^2_k = p''_{j(1)}q_{j(1)} \cdots p_{j(k)}q_{j(k)}p_{j(k)+1}$. Every couple q_i and $q_{j(k)}$ are associated by $\oplus$ for all $1 \leq i \leq k$.

Induction: Let $l = k+1$; $w_{k+1} = p_1q_1 \ldots p_{j(k)+2}q_{j(k)+2}$ $p_{j(k)+3}$, consequently. Let the formulae of the substrings $w_{k+1} = w^1_{k+1}w^2_{k+1}$ be $t^1_{k+1} \vdash m_1\mathbf{M}_1$ and $t^2_{k+1} \vdash m_2\mathbf{M}_2$, respectively. We can rewrite the substrings $w_{k+1} = w^1_{k+1}w^2_{k+1}$ in terms of $w_k = w^1_kw^2_k$ in three cases.

Case I: Suppose $w^1_{k+1} = pqw^1_k$. It follows that the direction of q is $<$. Since w^1_{k+1} combines w^2_{k+1}, we can conclude that $w^2_{k+1} = p'q'w^2_k$. Therefore, q and q' are also associated by $\oplus$.

Case II: Suppose $w^1_{k+1} = w^1_kqp$. It follows that the direction of q is $>$. Since w^1_{k+1} combines w^2_{k+1}, we can conclude that $w^2_{k+1} = w^2_kq'p'$. Therefore, q and q' are also associated by $\oplus$.

Case III: $w^1_{k+1} = p_1q_1 \ldots p_mq_mpqp_{m+1}q_{m+1} \ldots p_nq_np_{k+1}$ and $w^2_{k+1} = p_{j(1)}q_{j(1)} \cdots p_{j(m')}q_{j(m')}p'q'p_{j(m')+1}q_{j(m')+1} \cdots p_{j(k)}$ $q_{j(k)}p_{j(k)+1}$, where $1 < m,m' < k$. Since w^1_{k+1} and w^2_{k+1} combine and every q_i and $q_{j(i)}$ are associated, we can conclude that $m = m'$. Therefore, q and q' are also associated by $\oplus$.

From Case I, Case II, and Case III, we can rewrite $w^1_{k+1} = p'_1q'_1p'_2q'_2 \cdots p'_{k+1}$ and $w^2_{k+1} = p'_{j(1)}q'_{j(1)}p'_{j(2)}q'_{j(2)} \cdots p'_{j(k+1)}$. Since each q_i in w^1_k and $q_{j(i)}$ in w^2_k are already associated by $\oplus$, it follows that all q_i and $q_{j(i)+1}$ are also associated. ■

Dependency Parsing with Short Dependency Relations in Unlabeled Data

Wenliang Chen, Daisuke Kawahara, Kiyotaka Uchimoto, Yujie Zhang, Hitoshi Isahara
Computational Linguistics Group
National Institute of Information and Communications Technology
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto, Japan, 619-0289
{chenwl, dk, uchimoto, yujie, isahara}@nict.go.jp

Abstract

This paper presents an effective dependency parsing approach of incorporating short dependency information from unlabeled data. The unlabeled data is automatically parsed by a deterministic dependency parser, which can provide relatively high performance for short dependencies between words. We then train another parser which uses the information on short dependency relations extracted from the output of the first parser. Our proposed approach achieves an unlabeled attachment score of 86.52, an absolute 1.24% improvement over the baseline system on the data set of Chinese Treebank.

1 Introduction

In dependency parsing, we attempt to build the dependency links between words from a sentence. Given sufficient labeled data, there are several supervised learning methods for training high-performance dependency parsers(Nivre et al., 2007). However, current statistical dependency parsers provide worse results if the dependency length becomes longer (McDonald and Nivre, 2007). Here the length of a dependency from word w_i and word w_j is simply equal to $|i - j|$. Figure 1 shows the F_1 score[1] provided by a deterministic parser relative to dependency length on our testing data. From the figure, we find that F_1 score decreases when dependency length increases as (McDonald and Nivre, 2007) found. We also notice that the parser provides good results for short dependencies (94.57% for dependency length = 1 and 89.40% for dependency length = 2). In this paper, short dependency refers to the dependencies whose length is 1 or 2.

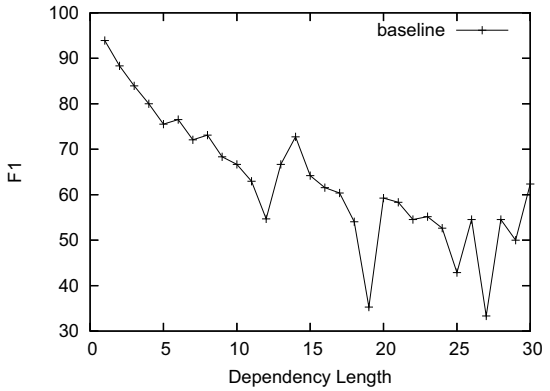


Figure 1: F-score relative to dependency length

Labeled data is expensive, while unlabeled data can be obtained easily. In this paper, we present an approach of incorporating unlabeled data for dependency parsing. First, all the sentences in unlabeled data are parsed by a dependency parser, which can provide state-of-the-art performance. We then extract information on short dependency relations from the parsed data, because the performance for short dependencies is relatively higher than others. Finally, we train another parser by using the information as features.

The proposed method can be regarded as a semi-supervised learning method. Currently, most semi-

[1] precision represents the percentage of predicted arcs of length d that are correct and recall measures the percentage of gold standard arcs of length d that are correctly predicted.
$F_1 = 2 \times precision \times recall/(precision + recall)$

supervised methods seem to do well with artificially restricted labeled data, but they are unable to outperform the best supervised baseline when more labeled data is added. In our experiments, we show that our approach significantly outperforms a state-of-the-art parser, which is trained on full labeled data.

2 Motivation and previous work

The goal in dependency parsing is to tag dependency links that show the head-modifier relations between words. A simple example is in Figure 2, where the link between *a* and *bird* denotes that *a* is the dependent of the head *bird*.

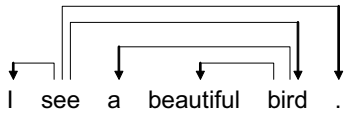


Figure 2: Example dependency graph.

We define that ***word distance*** of word w_i and word w_j is equal to $|i-j|$. Usually, the two words in a head-dependent relation in one sentence can be adjacent words (word distance = 1) or neighboring words (word distance = 2) in other sentences. For example, "a" and "bird" has head-dependent relation in the sentence at Figure 2. They can also be adjacent words in the sentence "I see a bird.".

Suppose that our task is Chinese dependency parsing. Here, the string "专家级JJ(Specialist-level)/工　作NN(working)/会　谈NN(discussion)" should be tagged as the solution (a) in Figure 3. However, our current parser may choose the solution (b) in Figure 3 without any additional information. The point is how to assign the head for "专家级(Specialist-level)". Is it "工作(working)" or "会谈(discussion)"?

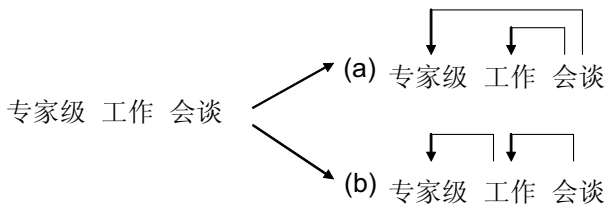


Figure 3: Two solutions for "专家级(Specialist-level)/工作(working)/会谈(discussion)"

As Figure 1 suggests, the current dependency parser is good at tagging the relation between adjacent words. Thus, we expect that dependencies of adjacent words can provide useful information for parsing words, whose word distances are longer. When we search the string "专家级(Specialist-level)/会谈(discussion)" at google.com, many relevant documents can be retrieved. If we have a good parser, we may assign the relations between the two words in the retrieved documents as Figure 4 shows. We can find that "会谈(discussion)" is the head of "专家级(Specialist-level)" in many cases.

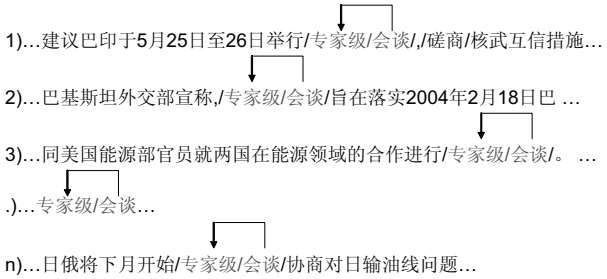


Figure 4: Parsing "专家级(Specialist-level)/会谈(discussion)" in unlabeled data

Now, consider what a learning model could do to assign the appropriate relation between "专家级(Specialist-level)" and "会谈(discussion)" in the string "专家级(Specialist-level)/工作(working)/会谈(discussion)". In this case, we provide additional information to "会谈(discussion)" as the possible head of "专家级(Specialist-level)" in the unlabeled data. In this way, the learning model may use this information to make correct decision.

Till now, we demonstrate how to use the dependency relation between adjacent words in unlabeled data to help tag the relation between two words whose word distance is 2. In the similar way, we can also assign the relation between two words whose word distance is longer by using the information.

Based on the above observations, we propose an approach of exploiting the information from a large-scale unlabeled data for dependency parsing. We use a parser to parse the sentences in unlabeled data. Then another parser makes use of the information on short dependency relations in the newly parsed data to improve performance.

Our study is relative to incorporating unlabeled

data into a model for parsing. There are several other studies relevant to ours as described below.

A simple method is self-training in which the existing model first labels unlabeled data and then the newly labeled data is then treated as hand annotated data for training a new model. But it seems that self-training is not so effective. (Steedman et al., 2003) reports minor improvement by using self-training for syntactic parsing on small labeled data. The reason may be that errors in the original model would be amplified in the new model. (McClosky et al., 2006) presents a successful instance of parsing with self-training by using a re-ranker. As Figure 1 suggests, the dependency parser performs bad for parsing the words with long distances. In our approach, we choose partial reliable information which comes from short dependency relations for the dependency parser.

(Smith and Eisner, 2006) presents an approach to improve the accuracy of a dependency grammar induction models by EM from unlabeled data. They obtain consistent improvements by penalizing dependencies between two words that are farther apart in the string.

The study most relevant to ours is done by (Kawahara and Kurohashi, 2006). They present an integrated probabilistic model for Japanese parsing. They also use partial information after current parser parses the sentences. Our work differs in that we consider general dependency relations while they only consider case frames. And we represent additional information as the features for learning models while they use the case frames as one component for a probabilistic model.

3 Our Approach

In this section, we describe our approach of exploiting reliable features from unlabeled data, which is parsed by a basic parser. We then train another parser based on new feature space.

3.1 Training a basic parser

In this paper, we implement a deterministic parser based on the model described by (Nivre, 2003). This model is simple and works very well in the shared-tasks of CoNLL2006(Nivre et al., 2006) and CoNLL2007(Hall et al., 2007). In fact, our approach can also be applied to other parsers, such as (Yamada and Matsumoto, 2003)'s parser, (McDonald et al., 2006)'s parser, and so on.

3.1.1 The parser

The parser predicts unlabeled directed dependencies between words in sentences. The algorithm (Nivre, 2003) makes a dependency parsing tree in one left-to-right pass over the input, and uses a stack to store the processed tokens. The behaviors of the parser are defined by four elementary actions (where TOP is the token on top of the stack and NEXT is the next token in the original input string):

- Left-Arc(LA): Add an arc from NEXT to TOP; pop the stack.
- Right-Arc(RA): Add an arc from TOP to NEXT; push NEXT onto the stack.
- Reduce(RE): Pop the stack.
- Shift(SH): Push NEXT onto the stack.

The first two actions mean that there is a dependency relation between TOP and NEXT.

More information about the parser can be available in the paper(Nivre, 2003). The parser uses a classifier to produce a sequence of actions for a sentence. In our experiments, we use the SVM model as the classifier. More specifically, our parser uses LIBSVM(Chang and Lin, 2001) with a polynomial kernel (degree = 3) and the built-in one-versus-all strategy for multi-class classification.

3.1.2 Basic features

We represent basic features extracted from the fields of data representation, including word and part-of-speech(POS) tags. The basic features used in our parser are listed as follows:

- The features based on words: the words of TOP and NEXT, the word of the head of TOP, the words of leftmost and rightmost dependent of TOP, and the word of the token immediately after NEXT in original input string.
- The features based on POS: the POS of TOP and NEXT, the POS of the token immediately below TOP, the POS of leftmost and rightmost dependent of TOP, the POS of next three tokens after NEXT, and the POS of the token immediately before NEXT in original input string.

With these basic features, we can train a state-of-the-art supervised parser on labeled data. In the following content, we call this parser Basic Parser.

3.2 Unlabeled data preprocessing and parsing

The input of our approach is unlabeled data, which can be obtained easily. For the Basic Parser, the corpus should have part-of-speech (POS) tags. Therefore, we should assign the POS tags using a POS tagger. For Chinese sentences, we should segment the sentences into words before POS tagging. After data preprocessing, we have the word-segmented sentences with POS tags. We then use the Basic Parser to parse all sentences in unlabeled data.

3.3 Using short dependency relations as features

The Basic Parser can provide complete dependency parsing trees for all sentences in unlabeled data. As Figure 1 shows, short dependencies are more reliable. To offer reliable information for the model, we propose the features based on short dependency relations from the newly parsed data.

3.3.1 Collecting reliable information

In a parsed sentence, if the dependency length of two words is 1 or 2, we add this word pair into a list named DepList and count its frequency. We consider the direction and length of the dependency. D1 refers to the pairs with dependency length 1, D2 refers to the pairs with dependency length 2, R refers to right arc, and L refers to left arc. For example, "专家级(specialist-level)" and "会谈(discussion)" are adjacent words in a sentence "我们(We)/举行(held)/专家级(specialist-level)/会谈(discussion)/。" and have a left dependency arc assigned by the Basic Parser. We add a word pair "专家级(specialist-level)-会谈(discussion)" with "D1-L" and its frequency into the DepList.

According to frequency, we then group word pairs into different buckets, with a bucket ONE for frequency 1, a single bucket LOW for 2-7, a single bucket MID for 8-14, and a single bucket HIGH for 15+. We choose these threshold values via testing on development data. For example, the frequency of the pair "专家级(specialist-level)-会谈(discussion)" with "D1-L" is 20. Then it is grouped into the bucket "D1-L-HIGH".

Here, we do not use the frequencies as the weight of the features. We derive the weights of the features by the SVM model from training data rather than approximating the weights from unlabeled data.

3.3.2 New features

Based on the DepList, we represent new features for training or parsing current two words: TOP and NEXT. We consider word pairs from the context around TOP and NEXT, and get the buckets of the pairs in the DepList.

First, we represent the features based on D1. We name these features D1 features. The D1 features are listed according to different word distances between TOP and NEXT as follows:

1. Word distance is 1: (TN0) the bucket of the word pair of TOP and NEXT, and (TN1) the bucket of the word pair of TOP and next token after NEXT.
2. Word distance is 2 or 3+: (TN0) the bucket of the word pair of TOP and NEXT, (TN1) the bucket of the word pair of TOP and next token after NEXT, and (TN_1) the bucket of the word pair of TOP and the token immediately before NEXT.

In item 2), all features are in turn combined with two sets of distances: a set for distance 2 and a single set for distances 3+. Thus, we have 8 types of D1 features, including 2 types in item 1) and 6 types in item 2). The feature is formatted as "Position:WordDistance:PairBucket". For example, we have the string "专家级(specialist-level)/w_1/w_2/w_3/会谈(discussion)", and "专家级(specialist-level)" is TOP and "会谈(discussion)" is NEXT. Thus we can have the feature "TN0:3+:D1-L-HIGH" for TOP and NEXT, because the word distance is 4(3+) and "专家级(specialist-level)-会谈(discussion)" belongs to the bucket "D1-L-HIGH". Here, if a string belongs to two buckets, we use the most frequent bucket.

Then, we represent the features based on D2. We name these features D2 features. The D2 features are listed as follows:

1. Word distance is 1: (TN1) the bucket of the word pair of TOP and next token after NEXT.

2. Word distance is 2: (TN0) the bucket of the word pair of TOP and NEXT, and (TN1) the bucket of the word pair of TOP and next token after NEXT.

4 Experiments

For labeled data, we used the Chinese Treebank (CTB) version 4.0[2] in our experiments. We used the same rules for conversion and created the same data split as (Wang et al., 2007): files 1-270 and 400-931 as training, 271-300 as testing and files 301-325 as development. We used the gold standard segmentation and POS tags in the CTB.

For unlabeled data, we used the PFR corpus [3]. It includes the documents from People's Daily at 1998 (12 months). There are about 290 thousand sentences and 15 million words in the PFR corpus. To simplify, we used its segmentation. And we discarded the POS tags because PFR and CTB used different POS sets. We used the package TNT (Brants, 2000), a very efficient statistical part-of-speech tagger, to train a POS tagger[4] on training data of the CTB.

We measured the quality of the parser by the unlabeled attachment score (UAS), i.e., the percentage of tokens with correct HEAD. We reported two types of scores: "UAS without p" is the UAS score without all punctuation tokens and "UAS with p" is the one with all punctuation tokens.

4.1 Experimental results

In the experiments, we trained the parsers on training data and tuned the parameters on development data. In the following sessions, "baseline" refers to Basic Parser (the model with basic features), and "OURS" refers to our proposed parser (the model with all features).

4.1.1 Our approach

Table 1 shows the results of the parser with different feature sets, where "+D1" refers to the parser with basic features and D1 features, and "+D2" refers to the parser with all features(basic features, D1 features, and D2 features). From the table, we found a large improvement (1.12% for UAS without p and 1.23% for UAS with p) from adding D1 features. And D2 features provided minor improvement, 0.12% for UAS without p and 0.14% for UAS with p. This may be due to the information from dependency length 2 containing more noise. Totally, we achieved 1.24% improvement for UAS with p and 1.37% for UAS without p. The improvement is significant in one-tail paired t-test ($p < 10^{-5}$).

Table 1: The results with different feature sets

	UAS without p	UAS with p
baseline	85.28	83.79
+D1	86.40	85.02
+D2(OURS)	86.52	85.16

We also attempted to discover the effect of different numbers of unlabeled sentences to use. Table 2 shows the results with different numbers of sentences. Here, we randomly chose different percentages of sentences from unlabeled data. When we used 1% sentences of unlabeled data, the parser achieved a large improvement. As we added more sentences, the parser obtained more benefit.

Table 2: The results with different numbers of unlabeled sentences

Sentences	UAS without p	UAS with p
0%(baseline)	85.28	83.79
1%	85.68	84.40
2%	85.69	84.51
5%	85.78	84.59
10%	85.97	84.62
20%	86.25	84.86
50%	86.34	84.92
100%(OURS)	86.52	85.16

4.1.2 Comparison of other systems

Finally, we compare our parser to the state of the art. We used the same testing data as (Wang et al., 2005) did, selecting the sentences length up to 40. Table 3 shows the results achieved by our method and other researchers (UAS with p), where Wang05 refers to (Wang et al., 2005), Wang07 refers

[2]More detailed information can be found at http://www.cis.upenn.edu/~chinese/.

[3]More detailed information can be found at http://www.icl.pku.edu.

[4]To know whether our POS tagger is good, we also tested the TNT package on the standard training and testing sets for full parsing (Wang et al., 2006). The TNT-based tagger provided 91.52% accuracy, the comparative result with (Wang et al., 2006).

to (Wang et al., 2007), and McDonald&Pereira06[5] refers to (McDonald and Pereira, 2006). From the table, we found that our parser performed best.

Table 3: The results on the sentences length up to 40

	UAS with p
Wang05	79.9
McDonald&Pereira06	82.5
Wang07	86.6
baseline	87.1
OURS	88.4

5 Analysis

5.1 Improvement relative to dependency length

We now look at the improvement relative to dependency length as Figure 5 shows. From the figure, we found that our method provided better performance when dependency lengths are less than 13. Especially, we had improvements 2.35% for dependency length 4, 3.13% for length 5, 2.56% for length 6, and 4.90% for length 7. For longer ones, the parser can not provide stable improvement. The reason may be that shorter dependencies are often modifier of nouns such as determiners or adjectives or pronouns modifying their direct neighbors, while longer dependencies typically represent modifiers of the root or the main verb in a sentence(McDonald and Nivre, 2007). We did not provide new features for modifiers of the root.

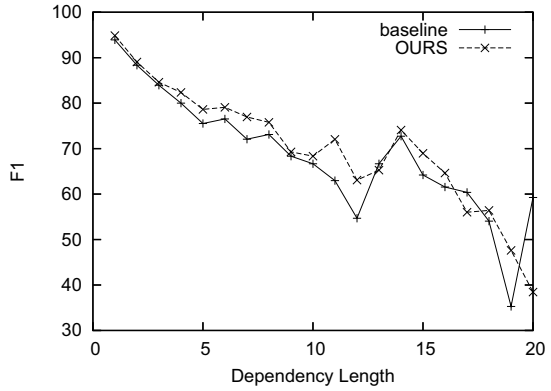


Figure 5: Improvement relative to dependency length

[5](Wang, 2007) reported this result.

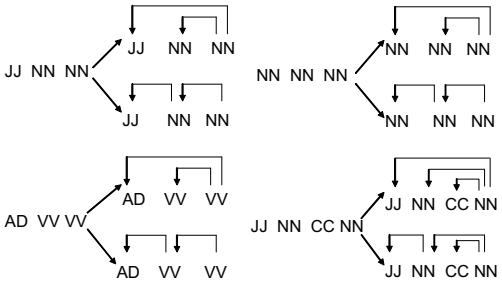


Figure 6: Ambiguities

5.2 Cases study in neighborhood

In Chinese dependency parsing, there are many ambiguities in neighborhood, such as "JJ NN NN", "AD VV VV", "NN NN NN", "JJ NN CC NN". They have possible parsing trees as Figure 6 shows. For these ambiguities, our approach can provide additional information for the parser. For example, we have the following case in the data set: "友好JJ(friendly)/ 合作NN(corporation)/ 关系NN(relationship)/". We can provide additional information about the relations of "友好JJ(friendly)/ 合作NN(corporation)" and "友好JJ(friendly)/ 关系NN(relationship)/" in unlabeled data to help the parser make the correct decision.

Our approach can also work for the longer constructions, such as "JJ NN NN NN" and "NN NN NN NN" in the similar way.

For the construction "JJ NN1 CC NN2", we now do not define special features to solve the ambiguity. However, based on the current DepList, we can also provide additional information about the relations of JJ/NN1 and JJ/NN2. For example, for the string "进一步JJ(further)/ 改善NN(improvement)/ 和CC(and)/ 发展NN(development)/", the parser often assigns "改善(improvement)" as the head of "进一步(further)" instead of "发展(development)". There is an entry "进一步(further)-发展(development)" in the DepList. Here, we need a coordination identifier to identify these constructions. After that, we can provide the information for the model.

6 Conclusion

This paper presents an effective approach to improve dependency parsing by using unlabeled data. We extract the information on short dependency relations

in an automatically generated corpus parsed by a basic parser. We then train a new parser with the information. The new parser achieves an absolute improvement of 1.24% over the state-of-the-art parser on Chinese Treebank (from 85.28% to 86.52%).

There are many ways in which this research should be continued. First, feature representation needs to be improved. Here, we use a simple feature representation on short dependency relations. We may use a combined representation to use the information from long dependency relations even they are not so reliable. Second, we can try to select more accurately parsed sentences. Then we may collect more reliable information than the current one.

References

T. Brants. 2000. TnT–a statistical part-of-speech tagger. *Proceedings of the 6th Conference on Applied Natural Language Processing*, pages 224–231.

C.C. Chang and C.J. Lin. 2001. LIBSVM: a library for support vector machines. *Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm*, 80:604–611.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryigit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? a study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 933–939.

D. Kawahara and S. Kurohashi. 2006. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 176–183.

D. McClosky, E. Charniak, and M. Johnson. 2006. Reranking and self-training for parser adaptation. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 337–344.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131.

R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of the 11th Conf. of the European Chapter of the ACL (EACL)*.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 216–220, New York City, June. Association for Computational Linguistics.

J. Nivre, J. Hall, J. Nilsson, G. Eryigit, and S Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

J. Nivre. 2003. An efficient algorithm for projective dependency parsing. *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.

Noah A. Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 569–576, Sydney, Australia, July. Association for Computational Linguistics.

M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. *The Proceedings of the Annual Meeting of the European Chapter of the ACL*, pages 331–338.

Qin Iris Wang, Dale Schuurmans, and Dekang Lin. 2005. Strictly lexical dependency parsing. In *IWPT2005*.

Mengqiu Wang, Kenji Sagae, and Teruko Mitamura. 2006. A Fast, Accurate Deterministic Parser for Chinese. In *Coling-ACL2006*.

Qin Iris Wang, Dekang Lin, and Dale Schuurmans. 2007. Simple training of dependency parsers via structured boosting. In *IJCAI2007*.

Qin Iris Wang. 2007. Learning structured classifiers for statistical dependency parsing. In *NAACL-HLT 2007 Doctoral Consortium*.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proc. of the 8th Intern. Workshop on Parsing Technologies (IWPT)*, pages 195–206.

An Effective Compositional Model for Lexical Alignment

Béatrice Daille **Emmanuel Morin**
Université de Nantes, LINA - FRE CNRS 2729
2, rue de la Houssinière, BP 92208
F-44322 Nantes cedex 03
{beatrice.daille,emmanuel.morin}@univ-nantes.fr

Abstract

The automatic compilation of bilingual dictionaries from comparable corpora has been successful for single-word terms (SWTs), but remains disappointing for multi-word terms (MWTs). One of the main problems is the insufficient coverage of the bilingual dictionary. Using the compositional translation method improved the results, but still shows some limits for MWTs of different syntactic structures. In this paper, we propose to bridge the gap between syntactic structures through morphological links. The results show a significant improvement in the compositional translation of MWTs that demonstrate the efficiency of the morphologically based-method for lexical alignment.

1 Introduction

Current research in the automatic compilation of bilingual dictionaries from corpora uses of comparable corpora. Comparable corpora gather texts sharing common features (domain, topic, genre, discourse) without having a source text-target text relationship. They are considered by human translators more trustworthy than parallel corpora (Bowker and Pearson, 2002). Moreover, they are available for any written languages and not only for pairs of languages involving English. The compilation of specialized dictionaries should take into account multi-word terms (MWTs) that are more precise and specific to a particular scientific domain than single-word terms (SWTs). The standard approach is based on lexical context analysis and relies on the simple observation that a SWT or a MWT and its translation tend to appear in the same lexical contexts. Correct results are obtained for SWTs with an accuracy of about 80% for the top 10-20 proposed candidates using large comparable corpora (Fung, 1998; Rapp, 1999; Chiao and Zweigenbaum, 2002) or 60% using small comparable corpora (Déjean and Gaussier, 2002). In comparison, the results obtained for MWTs are disappointing. For instance, (Morin et al., 2007) have achieved 30% and 42% precision for the top 10 and top 20 candidates in a 0.84 million-word French-Japanese corpus. These results could be explained by the low frequency of MWTs compared to SWTs, by the lack of parallelism between the source and the target MWT extraction systems, and by the low performance of the alignment program. For SWTs, the process is in two steps: looking in a dictionary, and if no direct translation is available, starting the contextual analysis. Looking in the dictionary gives low results for MWTs: 1% compared to 30% for French and 20% for Japanese SWTs (Morin and Daille, 2006). To extend the coverage of the bilingual dictionary, an intermediate step is added between looking in the dictionary and the contextual analysis that will propose several translation candidates to compare with the target MWTs. These candidate translations are obtained thanks to a compositional translation method (Melamed, 1997; Grefenstette, 1999). This method reveals some limits when MWTs in the source and the target languages do not share the same syntactic patterns.

In this paper, we put forward an extended compo-

sitional method that bridges the gap between MWTs of different syntactic structures through morphological links. We experiment within this method of French-Japanese lexical alignment, using multilingual terminology mining chain made up of two terminology extraction systems; one in each language, and an alignment program. The term extraction systems are publicly available and both extract MWTs. The alignment program makes use of the direct context-vector approach (Fung, 1998; Rapp, 1999). The results show an improvement of 33% in the translation of MWTs that demonstrate the efficiency of the morphologically based-method for lexical alignment.

2 Multilingual terminology mining chain

Taking a comparable corpora as input, the multilingual terminology mining chain outputs a list of single- and multi-word candidate terms along with their candidate translations (see Figure 1). This chain performs a contextual analysis that adapts the direct context-vector approach (Rapp, 1995; Fung and McKeown, 1997) for SWTs to MWTs. It consists of the following five steps:

1. For each language, the documents are cleaned, tokenized, tagged and lemmatized. For French, Brill's POS tagger[1] and the FLEM lemmatiser[2] are used, and for Japanese, ChaSen[3]. We then extract the MWTs and their variations using the ACABIT terminology extraction system available for French[4] (Daille, 2003), English and Japanese[5] (Takeuchi et al., 2004). (From now on, we will refer to lexical units as words, SWTs or MWTs).

2. We collect all the lexical units in the context of each lexical unit i and count their occurrence frequency in a window of n words around i. For each lexical unit i of the source and the target languages, we obtain a context vector v_i which gathers the set of co-occurrence units j associated with the number of times that j and i occur together occ^i_j. In order to identify specific words in the lexical context and to reduce word-frequency effects, we normalize context vectors using an association score such as Mutual Information (Fano, 1961) or Log-likelihood (Dunning, 1993).

3. Using a bilingual dictionary, we translate the lexical units of the source context vector. If the bilingual dictionary provides several translations for a lexical unit, we consider all of them but weigh the different translations by their frequency in the target language.

4. For a lexical unit to be translated, we compute the similarity between the translated context vector and all target vectors through vector distance measures such as Cosine (Salton and Lesk, 1968) or Jaccard (Tanimoto, 1958).

5. The candidate translations of a lexical unit are the target lexical units closest to the translated context vector according to vector distance.

In this approach, the translation of the lexical units of the context vectors (step 3 of the previous approach), which depends on the coverage of the bilingual dictionary vis-à-vis the corpus, is the most important step: the greater the number of elements translated in the context vector, the more discriminating the context vector in selecting translations in the target language. Since the lexical units refer to SWTs and MWTs, the dictionary must contain many entries which occur in the corpus. For SWTs, combining a general bilingual dictionary with a specialized bilingual dictionary or a multilingual thesaurus to translate context vectors ensures that much of their elements will be translated (Chiao and Zweigenbaum, 2002; Déjean et al., 2002). For a MWT to be translated, steps 3 to 5 could be avoided thanks to a compositional method that will propose several translation candidates to directly compare with the target MWTs identified in step 1. Moreover, the compositional method is useful in step 3 to compensate for the bilingual dictionary when the multi-word units of the context vector are not directly translated.

[1] http://www.atilf.fr/winbrill/

[2] http://www.univ-nancy2.fr/pers/namer/

[3] http://chasen-legacy.sourceforge.jp/

[4] http://www.sciences.univ-nantes.fr/info/perso/permanents/daille/ and release for Mandriva Linux.

[5] http://cl.cs.okayama-u.ac.jp/rsc/jacabit/

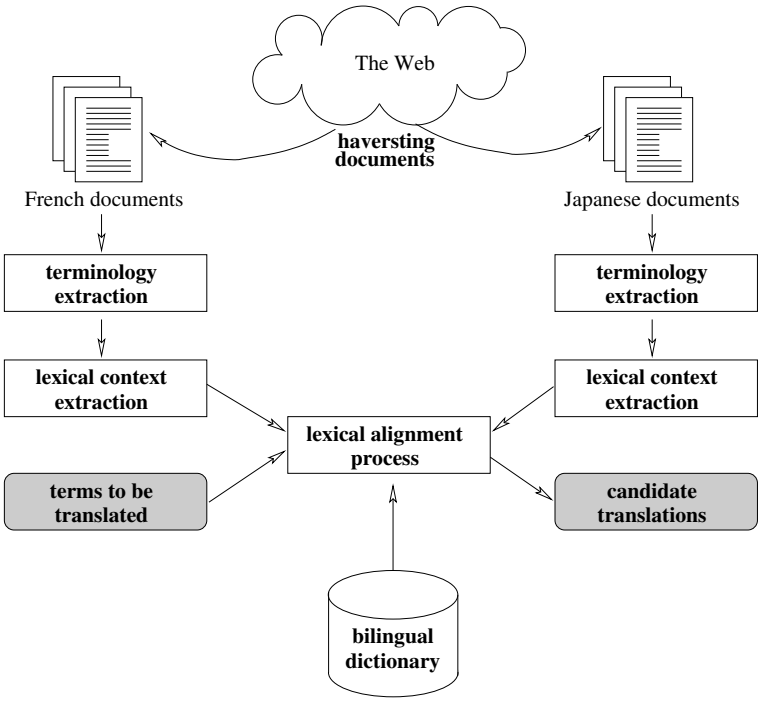


Figure 1: Architecture of the multilingual terminology mining chain

3 Default compositional method

In order to increase the coverage of the dictionary for MWTs that could not be directly translated, we generated possible translations by using a default compositional method (Melamed, 1997; Grefenstette, 1999).

For each element of the MWT found in the bilingual dictionary, we generated all the translated combinations identified by the terminology extraction system. For example, for the French MWT *fatigue chronique* (*chronic fatigue*), there are four Japanese translations for *fatigue* (*fatigue*) – 疲れ, 疲労, 倦怠, 飽き – and two translations for *chronique* (*chronic*) – 記事番組, 慢性. Next, we generated all possible combinations of the translated elements (see Table 1[6]) and selected those which refer to an existing MWT in the target language. In the above example, only one term for each element was identified by the Japanese extraction system: 慢性疲労. In this approach, when it is not possible to translate all parts of an MWT, or when the translated combinations are not identified by the extraction system, the MWT is not taken into account in the translation step.

chronique	*fatigue*
記事番組	疲れ
慢性	疲れ
記事番組	疲労
慢性	疲労
記事番組	倦怠
慢性	倦怠
記事番組	飽き
慢性	飽き

Table 1: Illustration of the compositional method (the underlined Japanese MWT actually exists)

This approach also differs from that used by (Robitaille et al., 2006) for French-Japanese translation. They first decompose the French MWT into combinations of shorter multi-word unit elements. This approach makes the direct translation of a subpart of the MWT possible if it is present in the bilingual dictionary. For MWTs of length n, (Robitaille et al., 2006) produce all the combinations of shorter multi-word unit elements of a length less than or equal to n. For

[6]The French word order is reversed to take into account the different constraints between French and Japanese.

example, the French MWT *syndrome de fatigue chronique* (*chronic fatigue disorder*) yields the following four combinations: i) [*syndrome de fatigue chronique*], ii) [*syndrome de fatigue*] [*chronique*], iii) [*syndrome*] [*fatigue chronique*] and iv) [*syndrome*] [*fatigue*] [*chronique*]. We limit ourselves to the combination of type iv) above since 90% of the French candidate terms provided by the term extraction process after clustering are only composed of two content words.

4 Pattern switching

The compositional translation presents problems which have been reported by (Baldwin and Tanaka, 2004; Brown et al., 1993):

Fertility SWTs and MWTs are not translated by a term of a same length. For instance, the French SWT *hypertension* (*hypertension*) is translated by the Japanese MWT 高血圧 (here the kanji 高 (*taka*) means *high* and the term 血圧 (*ketsuatsu*) means *blood pressure*).

Pattern switching MWTs in the source and the target language do not share the same syntactic patterns. For instance, the French MWT *cellule graisseuse* (*fat cell*) of `N ADJ` structure is translated by the Japanese MWT 脂肪細胞 of `N N` structure where the French noun *cellule* is translated by the Japanese noun 細胞 (*saiboo - cellule - cell*) and the French adjective *graisseuse* by the Japanese noun 脂肪 (*shiboo - graisse - fat*).

Foreign name When a proper name is part of the MWT, it is not always translated: within the French MWT *syndrome de Cushing* (*Cushing syndrome*), Cushing is either transliterated クッシング症候群 or remains unchanged *Cushing*症候群. The foreign name *Cushing* is of course not present in the dictionary.

The pattern switching problem involves the Adjective/Noun and the Noun/Verb part-of-speech switches. The Adjective/Noun switch commonly involves a relational adjective (`ADJR`). According to grammatical tradition, there are two main categories among adjectives: epithetic adjectives such as *important* (*significant*) and relational adjectives such as *sanguin* (*blood*). The former cannot have an agentive interpretation in contrast to the latter: the adjective *sanguin* (*blood*) within the MWT *acidité sanguine* (*blood acidity*) is an argument to the predicative noun *acidité* (*acidity*) and this is not the case for the adjective *important* (*significant*) within the noun phrase *acidité importante* (*significant acidity*). Such adjectives hold a naming function (Levi, 1978) and are particularly frequent in scientific fields (Daille, 2001). Relational adjectives are either denominal adjectives, morphologically derived from a noun thanks to a suffix, or adjectives having a noun usage such as *mathématique* (*mathematical/mathematics*). For the former, there are appropriate adjective-forming suffixes for French that lead to relational adjectives such as *-ique*, *-aire*, *-al*. For a noun, it is not possible to guess the adjective-forming suffix that will be employed as well as the alternation of the noun stem that could occur. Relational adjectives part of a MWT are often translated by a noun whatever the target language is. From French to Japanese, the examples are numerous: *prescription médicamenteuse* (処方薬 - *medicinal prescription*), *surveillance glycémique* (血糖管理 - *glycemic monitoring*), *fibre alimentaire* (食物繊維 - *dietary fibre*), *produit laitier* (乳製品 - *dairy product*), *fonction rénale* (腎臓機能 - *kidney function*).

The problem of fertility could only be solved thanks to a contextual analysis in contrast to the foreign name problem that could be solved by an heuristic. We decided to concentrate on the MWT pattern switching problem.

5 Morphologically-based compositional method

When it is not possible to directly translate a MWT — i.e. i) before performing the steps 3 to 5 of the contextual analysis for a multi-word term to be translated or ii) during step 3 for the translation of multi-word units of the context vector —, we first try to translate the MWT using the default compositional method. If the default compositional method fails, we use a morphologically-based compositional method. For each MWT of `N ADJ` structure, we generate candidate MWTs of `N Prep N` structure thanks to the rewriting rule:

$$
\begin{aligned}
&\mathrm{N}_1\ \mathrm{ADJ} \rightarrow \mathrm{N}_1\ \mathrm{Prep}\ \mathrm{Art}^{?}\ \mathcal{M}(\mathrm{ADJ}, \mathrm{N}_2)\\
&\mathcal{M}(\mathrm{ADJ}, \mathrm{N}_2) = [-\mathrm{ique}, -\mathrm{ie}]\\
&\mathcal{M}(\mathrm{ADJ}, \mathrm{N}_2) = [-\mathrm{ulaire}, -\mathrm{le}]\\
&\mathcal{M}(\mathrm{ADJ}, \mathrm{N}_2) = [-\mathrm{seux},]\\
&\ldots
\end{aligned}
\tag{1}
$$

$\mathcal{M}(\mathrm{ADJ}, \mathrm{N}_2)$ gathers a relational adjective ADJ such as *glycém-ique* and the noun N_2 from which the adjective has been derived such as *glycém-ie* thanks to the stripping-recoding rule $[-\mathrm{ique}, -\mathrm{ie}]$. We generate all possible forms of N_2 as matching stripping-recoding rules and keep those that belong to the biligual dictionary such as *glycém-ie*. Thus, we have created a morphological link between the MWT *contrôle glycémique* (*glycemic control*) of `N ADJ` structure and multi-word unit (MWU) of `N Prep N` structure *contrôle de la glycémie* (lit. *control of glycemia*). Since it has not been possible to translate all the parts of the MWT *contrôle glycémique*, because *glycémique* was not found in the dictionary, we use the morpholocally-linked MWU *contrôle de la glycémie* of which all the parts are translated. The morpholocally-linked MWU could be seen as a canonical lexical form in the translation process that possibly does not exist in the source language. For instance, if *index glycémique* (*glycemic index*) is a French MWT, the MWU *index de la glycémie* (lit. *index of the glycemia*) does not appear in the French corpus.

The stripping-recoding rules could be manually encoded, mined from a monolingual corpus using a learning method such as (Mikheev, 1997), or supplied by a source terminology extraction system that handles morphological variations. For such a system, a MWT is a canonical form which merges several synonymic variations. For instance, the French MWT *excès pondéral* (*overweight*) is the canonical form of the following variants: *excès pondéral* (*overweight*) of `N ADJ` structure, *excès de poids* (*overweight*) of `N PREP N` structure. It is this last method that we used for our experiment.

6 Evaluation

In this section, we will outline the different linguistic resources used for our experiments. We then evaluate the performance of the default and morphologically-based compositional methods.

6.1 Linguistic resources

In order to obtain comparable corpora, we selected the French and Japanese documents from the Web. The documents were taken from the medical domain, within the sub-domain of 'diabetes' and 'nutrition'. Document harvesting was carried out by a domain-based search, then by manual selection. A search for documents sharing the same domain can be achieved using keywords reflecting the specialized domain: for French *alimentation*, *diabète* and *obésité* (*food*, *diabetes*, and *obesity*); for Japanese, 糖尿病 and 肥満 (*diabetes*, and *overweight*). Then the documents were manually selected by native speakers of each language who are not domain specialists. These documents (248 for French and 538 for Japanese) were converted into plain text from HTML or PDF, yielding 1.5 million-word corpus (0.7 million-word for French and 0.8 million-word for Japanese).

The French-Japanese bilingual dictionary used in the translation phase was composed of four dictionaries freely available on the Web ([dico 1][7], [dico 2][8], [dico 3][9], and [dico 4][10]), and the French-Japanese Scientific Dictionary (1989) (called [dico 5]). Besides [dico 4], which deals with the medical domain, the other resources are general (as [dico 1, 2, and 3]) or technical (as [dico 5]) dictionaries. Merging the dictionaries yields a single resource with 173,156 entries (114,461 single words and 58,695 multi words) and an average of 2.1 translations per entry.

6.2 French N ADJ reference lists

We needed to distinguish between relational and epithetic adjectives appearing among the French N ADJ candidates to demonstrate the relevance of the morphological links. To build two French `N ADJ` reference lists, we proceeded as follows:

1. From the list of MWT candidates, we selected those sharing a `N ADJ` structure.

2. We kept only the candidate terms which occur

[7] `http://kanji.free.fr/`

[8] `http://quebec-japon.com/lexique/index.php?a=index&d=25`

[9] `http://dico.fj.free.fr/index.php`

[10] `http://quebec-japon.com/lexique/index.php?a=index&d=3`

more than 2 twice in the French corpus. As a result of filtering, 1,999 candidate terms were extracted.

3. We manually selected linguistically well-formed candidate terms. Here, 360 candidate terms were removed that included: misspelled terms, English terms, or subparts of longer terms.

4. We took out the terms that are directly translated by the bilingual dictionary and found in the comparable corpora. We identified 61 terms of which 30 use a relational adjective such as *vaisseau sanguin* (*blood vessel* - 血管), *produit laitier* (*dairy product* - 乳製品) and *insuffisance cardiaque* (*heart failure* - 心不全).

Finally, we created two French reference lists:

- [N ADJE] composed of 749 terms where ADJE is a epithetic adjective;
- [N ADJR] composed of 829 terms where ADJR is a relational adjective.

6.3 Default compositional method

We first evaluated the quality of the default compositional method for the two French reference lists. Table 2 shows the results obtained. The first three columns indicate the number of French and Japanese terms found in the comparable corpora, and the number of correct French-Japanese translations.

The results of this experiment show that only a small quantity of terms were translated by the default compositional method. Here, the terms belonging to [N ADJE] were more easily translated (10% with a precision of 69%) than the terms belonging to [N ADJR] (1%). We were unable to generate any translations for 56 (12%) and 227 (27%) terms respectively from the [N ADJE] and [N ADJR] lists. This was because one or several content words of the MWT candidates were not present in the bilingual dictionary. The best translations of candidates belonging to the [N ADJE] list are those where the adjective refers to a quantity such as *faible* (*low*), *moyen* (*medium*), or *haut* (*high*). Since our French-Japanese dictionary contained a small quantity of medical terms, the identified translations of the candidates belonging to the [N ADJR] list refers to generic relational adjectives such as *poids normal* (*standard weight* - 正常体重), *étude nationale* (*national study* - 全国調査), or *activité physique* (*physical activity* - 身体活動). We noticed that some generated MWUs do not exist in French such as *poids (de) norme* (*standard weight*), only the N ADJR form exists.

	# French terms	# Japanese terms	# correct translations
[N ADJE]	76	98	68
[N ADJR]	8	8	5

Table 2: Production of the default compositional method

6.4 Morphologically-based compositional method

We will now turn to the evaluation of the morphologically-based compositional method is are dedicated to the translation of the [N ADJR] list (see Table 4).

By comparison with the previous method, the results of this experiment show that a significant quantity of terms have now been translated. Since the compositional method can yield several Japanese translations for one French term, we associated 170 Japanese terms to 128 French terms with a high level of precision: 88.2%. Here, we were unable to generate any translations for 136 (16%) terms in comparison with the 227 terms (27%) for the default compositional method.

	# French terms	# Japanese terms	# correct translations
[N ADJR]	128	170	150

Table 4: Production of the morphologically-based compositional method

In Table 3, each French suffix is associated with the number of identified translations. The most productive suffixes are *-ique* such as *glycémie*/*glycémique* (*glycemia*/*glycemic*), *-al* such as *rein*/*rénal* (*kidney*/*renal*), *-el* such as

Suffix	# occ.	French term	Japanese term	(English)
-ique	94	*patient diabétique*	糖尿病患者	(*diabetes patient*)
-al	27	*traitement hormonal*	ホルモン療法	(*hormonal therapy*)
-el	18	*trouble nutritionnel*	栄養障害	(*nutritional disorder*)
-aire	15	*cellule musculaire*	筋肉細胞	(*muscular cell*)
-if	5	*apport nutritif*	栄養摂取	(*nutrition intake*)
-euse	4	*cellule graisseuse*	脂肪細胞	(*fat cell*)
-ier	4	*centre hospitalier*	センター病院	(*hospital complex*)
-ien	2	*hormone thyroïdien*	甲状腺ホルモン	(*thyroid hormone*)
-in	1	*lipide sanguin*	血液脂質	(*blood lipid*)

Table 3: Production of relational adjective according to suffix

corps/*corporel* (*body*/*bodily*), and *-aire* such as *aliment*/*alimentaire* (*food*/*dietary*).

Finally from 859 terms relative to `N ADJR` structure, we translated 30 terms (5.1%) with the dictionary, 5 terms (0.6%) by the default compositional method, and 150 terms (17.5%) by the morphologically-based compositional method. It was difficult to find more translations for several reasons: i) some specialized adjectives or nouns were not included in our resources, ii) some terms were not taken into account by the Japanese extraction system, and iii) some terms were not included in the Japanese corpus.

7 Conclusion and future work

This study investigated the compilation of bilingual terminologies from comparable corpora and showed how to push back the limits of the methods used in alignment programs to translate both single and multi- word terms. We proposed an extended compositional method that bridges the gap between MWTs of different syntactic structures through morphological links. We experimented with the method on MWTs of `N ADJ` structure involving a relational adjective. By the use of a list of stripping-recoding rules conjugated with a terminology extraction system, the method was more efficient than the default compositional method. The evaluation proposed at the end of the paper shows that 170 French-Japanese MWTs were extracted with a high precision (88.2%). This increases the coverage of the French-Japanese terminology of MWTs that can be obtained by the bilingual dictionary or the default compositional method. We are aware that the efficiency of this method relies on the completeness of the morphological ressources, dictionaries and stripping-recoding rules. Such resources need to be up todate for new domains and corpus.

In this study, we have observed that MWTs are of a different nature in each language: French patterns cover nominal phrases while Japanese patterns focus on morphologically-built compounds. A Japanese nominal phrase is not considered as a term: thus, the Japanese extraction system does not identify カロリーの摂取 (*caloric intake*) as a candidate MWT but カロリー摂取, unlike the French extraction system which does the contrary (*apport calorique* - *caloric intake*). Since our morphologically-based compositional method associated カロリー摂取 to *apport calorique*, we could yield the nominal phrase カロリーの摂取 and improve lexical alignment.

References

Timothy Baldwin and Takaaki Tanaka. 2004. Translation by Machine of Complex Nominals: Getting it Right. In *Proceedings of the ACL 2004 Workshop on Multiword Expressions: Integrating Processing*, pages 24–31, Barcelona, Spain.

Lynne Bowker and Jennifer Pearson. 2002. *Working with Specialized Language: A Practical Guide to Using Corpora*. Routeledge, London/New York.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Yun-Chuang Chiao and Pierre Zweigenbaum. 2002. Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pages 1208–1212, Tapei, Taiwan.

Béatrice Daille. 2001. Qualitative terminology extraction. In Didier Bourigault, Christian Jacquemin, and Marie-Claude L'Homme, editors, *Recent Advances in Computational Terminology*, volume 2 of *Natural Language Processing*, pages 149–166. John Benjamins.

Béatrice Daille. 2003. Terminology Mining. In Maria Teresa Pazienza, editor, *Information Extraction in the Web Era*, pages 29–44. Springer.

Hervé Déjean and Éric Gaussier. 2002. Une nouvelle approche à l'extraction de lexiques bilingues à partir de corpus comparables. *Lexicometrica, Alignement lexical dans les corpus multilingues*, pages 1–22.

Ted Dunning. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74.

Hervé Déjean, Fatia Sadat, and Éric Gaussier. 2002. An approach based on multilingual thesauri and model combination for bilingual lexicon extraction. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pages 218–224, Tapei, Taiwan.

Robert M. Fano. 1961. *Transmission of Information: A statistical Theory of Communications*. MIT Press, Cambridge, MA, USA.

French-Japanese Scientific Dictionary. 1989. Hakusuisha. 4th edition.

Pascale Fung and Kathleen McKeown. 1997. Finding Terminology Translations from Non-parallel Corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora (VLC'97)*, pages 192–202, Hong Kong, China.

Pascale Fung. 1998. A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-parallel Corpora. In David Farwell, Laurie Gerber, and Eduard Hovy, editors, *Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA'98)*, pages 1–16, Langhorne, PA, USA.

Gregory Grefenstette. 1999. The Word Wide Web as a Resource for Example-Based Machine Translation Tasks. In *ASLIB'99 Translating and the Computer 21*, London, UK.

Judith Levi. 1978. *The syntax and the semantics of complex nominals*. Academic Press, London.

I. Dan Melamed. 1997. A Word-to-Word Model of Translational Equivalence. In Philip R. Cohen and Wolfgang Wahlster, editors, *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 490–497, Madrid, Spain.

Andrei Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

Emmanuel Morin and Béatrice Daille. 2006. Comparabilité de corpus et fouille terminologique multilingue. *Traitement Automatique des Langues (TAL)*, 47(2):113–136.

Emmanuel Morin, Béatrice Daille, Koichi Takeuchi, and Kyo Kageura. 2007. Bilingual Terminology Mining – Using Brain, not brawn comparable corpora. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, pages 664–671, Prague, Czech Republic.

Reinhard Rapp. 1995. Identify Word Translations in Non-Parallel Texts. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'95)*, pages 320–322, Boston, MA, USA.

Reinhard Rapp. 1999. Automatic Identification of Word Translations from Unrelated English and German Corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pages 519–526, College Park, MD, USA.

Xavier Robitaille, Xavier Sasaki, Masatsugu Tonoike, Satoshi Sato, and Satoshi Utsuro. 2006. Compiling French-Japanese Terminologies from the Web. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, pages 225–232, Trento, Italy.

Gerard Salton and Michael E. Lesk. 1968. Computer evaluation of indexing and text processing. *Journal of the Association for Computational Machinery*, 15(1):8–36.

Koichi Takeuchi, Kyo Kageura, Béatrice Daille, and Laurent Romary. 2004. Construction of grammar based term extraction model for japanese. In Sophia Ananadiou and Pierre Zweigenbaum, editors, *Proceedings of the COLING 2004, 3rd International Workshop on Computational Terminology (COMPUTERM'04)*, pages 91–94, Geneva, Switzerland.

T. T. Tanimoto. 1958. An elementary mathematical theory of classification. Technical report, IBM Research.

Determining the Unithood of Word Sequences using a Probabilistic Approach

Wilson Wong, Wei Liu and Mohammed Bennamoun
School of Computer Science and Software Engineering
University of Western Australia
Crawley WA 6009
{wilson,wei,bennamou}@csse.uwa.edu.au

Abstract

Most research related to unithood were conducted as part of a larger effort for the determination of termhood. Consequently, novelties are rare in this small sub-field of term extraction. In addition, existing work were mostly empirically motivated and derived. We propose a new probabilistically-derived measure, independent of any influences of termhood, that provides dedicated measures to gather linguistic evidence from parsed text and statistical evidence from Google search engine for the measurement of unithood. Our comparative study using $1,825$ test cases against an existing empirically-derived function revealed an improvement in terms of precision, recall and accuracy.

1 Introduction

Automatic term recognition, also referred to as *term extraction* or *terminology mining*, is the process of extracting lexical units from text and filtering them for the purpose of identifying terms which characterise certain domains of interest. This process involves the determination of two factors: *unithood* and *termhood*. Unithood concerns with whether or not a sequence of words should be combined to form a more stable lexical unit. On the other hand, termhood measures the degree to which these stable lexical units are related to domain-specific concepts. Unithood is only relevant to *complex terms* (i.e. multi-word terms) while termhood (Wong et al., 2007a) deals with both *simple terms* (i.e. single-word terms) and complex terms. Recent reviews by (Wong et al., 2007b) show that existing research on unithood are mostly carried out as a prerequisite to the determination of termhood. As a result, there is only a small number of existing measures dedicated to determining unithood. Besides the lack of dedicated attention in this sub-field of term extraction, the existing measures are usually derived from term or document frequency, and are modified as per need. As such, the significance of the different weights that compose the measures usually assume an empirical viewpoint. Obviously, such methods are at most inspired by, but not derived from formal models (Kageura and Umino, 1996).

The three objectives of this paper are (1) to separate the measurement of unithood from the determination of termhood, (2) to devise a probabilistically-derived measure which requires only one threshold for determining the unithood of word sequences using non-static textual resources, and (3) to demonstrate the superior performance of the new probabilistically-derived measure against existing empirical measures. In regards to the first objective, we will derive our probabilistic measure free from any influence of termhood determination. Following this, our unithood measure will be an independent tool that is applicable not only to term extraction, but many other tasks in information extraction and text mining. Concerning the second objective, we will devise our new measure, known as the *Odds of Unithood* (OU), which are derived using Bayes Theorem and founded on a few elementary probabilities. The probabilities are estimated using Google page counts in an attempt to eliminate problems related to the use of static corpora. Moreover, only

one threshold, namely, OU_T is required to control the functioning of OU. Regarding the third objective, we will compare our new OU against an existing empirically-derived measure called *Unithood* (UH) (Wong et al., 2007b) in terms of their precision, recall and accuracy.

In Section 2, we provide a brief review on some of existing techniques for measuring unithood. In Section 3, we present our new probabilistic approach, the measures involved, and the theoretical and intuitive justification behind every aspect of our measures. In Section 4, we summarize some findings from our evaluations. Finally, we conclude this paper with an outlook to future work in Section 5.

2 Related Works

Some of the most common measures of unithood include pointwise *mutual information (MI)* (Church and Hanks, 1990) and *log-likelihood ratio* (Dunning, 1994). In mutual information, the co-occurrence frequencies of the constituents of complex terms are utilised to measure their dependency. The mutual information for two words a and b is defined as:

$$MI(a,b) = \log_2 \frac{p(a,b)}{p(a)p(b)} \tag{1}$$

where $p(a)$ and $p(b)$ are the probabilities of occurrence of a and b. Many measures that apply statistical techniques assuming strict normal distribution, and independence between the word occurrences (Franz, 1997) do not fare well. For handling extremely uncommon words or small sized corpus, *log-likelihood ratio* delivers the best precision (Kurz and Xu, 2002). Log-likelihood ratio attempts to quantify how much more likely one pair of words is to occur compared to the others. Despite its potential, *"How to apply this statistic measure to quantify structural dependency of a word sequence remains an interesting issue to explore."* (Kit, 2002). (Seretan et al., 2004) tested mutual information, log-likelihood ratio and t-tests to examine the use of results from web search engines for determining the collocational strength of word pairs. However, no performance results were presented.

(Wong et al., 2007b) presented a hybrid approach inspired by mutual information in Equation 1, and *C-value* in Equation 3. The authors employ Google page counts for the computation of statistical evidences to replace the use of frequencies obtained from static corpora. Using the page counts, the authors proposed a function known as *Unithood (UH)* for determining the mergeability of two lexical units a_x and a_y to produce a stable sequence of words s. The word sequences are organised as a set $W = \{s, a_x, a_y\}$ where $s = a_x b a_y$ is a term candidate, b can be any preposition, the coordinating conjunction *"and"* or an empty string, and a_x and a_y can either be noun phrases in the form Adj^*N+ or another s (i.e. defining a new s in terms of other s). The authors define UH as:

$$UH(a_x,a_y) = \begin{cases} 1 & \text{if } (MI(a_x,a_y) > MI^+) \vee \\ & (MI^+ \geq MI(a_x,a_y) \\ & \geq MI^- \wedge \\ & ID(a_x,s) \geq ID_T \wedge \\ & ID(a_y,s) \geq ID_T \wedge \\ & IDR^+ \geq IDR(a_x,a_y) \\ & \geq IDR^-) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where MI^+, MI^-, ID_T, IDR^+ and IDR^- are thresholds for determining mergeability decisions, and $MI(a_x, a_y)$ is the mutual information between a_x and a_y, while $ID(a_x, s)$, $ID(a_y, s)$ and $IDR(a_x, a_y)$ are measures of lexical independence of a_x and a_y from s. For brevity, let z be either a_x or a_y, and the independence measure $ID(z, s)$ is then defined as:

$$ID(z,s) = \begin{cases} \log_{10}(n_z - n_s) & \text{if}(n_z > n_s) \\ 0 & \text{otherwise} \end{cases}$$

where n_z and n_s is the Google page count for z and s respectively. On the other hand, $IDR(a_x, a_y) = \frac{ID(a_x,s)}{ID(a_y,s)}$. Intuitively, $UH(a_x, a_y)$ states that the two lexical units a_x and a_y can only be merged in two cases, namely, 1) if a_x and a_y has extremely high mutual information (i.e. higher than a certain threshold MI^+), or 2) if a_x and a_y achieve average mutual information (i.e. within the acceptable range of two thresholds MI^+ and MI^-) due to both of their extremely high independence (i.e. higher than the threshold ID_T) from s.

(Frantzi, 1997) proposed a measure known as *Cvalue* for extracting complex terms. The measure

is based upon the claim that a substring of a term candidate is a candidate itself given that it demonstrates adequate independence from the longer version it appears in. For example, *"E. coli food poisoning"*, *"E. coli"* and *"food poisoning"* are acceptable as valid complex term candidates. However, *"E. coli food"* is not. Therefore, some measures are required to gauge the strength of word combinations to decide whether two word sequences should be merged or not. Given a word sequence a to be examined for unithood, the *Cvalue* is defined as:

$$Cvalue(a) = \begin{cases} \log_2 |a| f_a & \text{if } |a| = g \\ \log_2 |a| (f_a - \frac{\sum_{l \in L_a} f_l}{|L_a|}) & \text{otherwise} \end{cases} \quad (3)$$

where $|a|$ is the number of words in a, L_a is the set of longer term candidates that contain a, g is the longest n-gram considered, f_a is the frequency of occurrence of a, and $a \notin L_a$. While certain researchers (Kit, 2002) consider *Cvalue* as a termhood measure, others (Nakagawa and Mori, 2002) accept it as a measure for unithood. One can observe that longer candidates tend to gain higher weights due to the inclusion of $\log_2 |a|$ in Equation 3. In addition, the weights computed using Equation 3 are purely dependent on the frequency of a.

3 A Probabilistically-derived Measure for Unithood Determination

We propose a probabilistically-derived measure for determining the unithood of word pairs (i.e. potential term candidates) extracted using the head-driven left-right filter (Wong, 2005; Wong et al., 2007b) and Stanford Parser (Klein and Manning, 2003). These word pairs will appear in the form of $(a_x, a_y) \in A$ with a_x and a_y located immediately next to each other (i.e. $x + 1 = y$), or separated by a preposition or coordinating conjunction *"and"* (i.e. $x + 2 = y$). Obviously, a_x has to appear before a_y in the sentence or in other words, $x < y$ for all pairs where x and y are the word offsets produced by the Stanford Parser. The pairs in A will remain as potential term candidates until their unithood have been examined. Once the unithood of the pairs in A have been determined, they will be referred to as *term candidates*. Formally, the unithood of any two lexical units a_x and a_y can be defined as

Definition 1 The unithood of two lexical units is the *"degree of strength or stability of syntagmatic combinations and collocations" (Kageura and Umino, 1996)* between them.

It is obvious that the problem of measuring the unithood of any pair of words is the determination of their *"degree"* of collocational strength as mentioned in Definition 1. In practical terms, the *"degree"* mentioned above will provide us with a way to determine if the units a_x and a_y should be combined to form s, or left alone as separate units. The collocational strength of a_x and a_y that exceeds a certain threshold will demonstrate to us that s is able to form a stable unit and hence, a better term candidate than a_x and a_y separated. It is worth pointing that the size (i.e. number of words) of a_x and a_y is not limited to 1. For example, we can have a_x="*National Institute*", b="*of*" and a_y="*Allergy and Infectious Diseases*". In addition, the size of a_x and a_y has no effect on the determination of their unithood using our approach.

As we have discussed in Section 2, most of the conventional practices employ frequency of occurrence from local corpora, and some statistical tests or information-theoretic measures to determine the coupling strength between elements in $W = \{s, a_x, a_y\}$. Two of the main problems associated with such approaches are:

- Data sparseness is a problem that is well-documented by many researchers (Keller et al., 2002). It is inherent to the use of local corpora that can lead to poor estimation of parameters or weights; and
- Assumption of independence and normality of word distribution are two of the many problems in language modelling (Franz, 1997). While the independence assumption reduces text to simply a bag of words, the assumption of normal distribution of words will often lead to incorrect conclusions during statistical tests.

As a general solution, we innovatively employ results from web search engines for use in a probabilistic framework for measuring unithood.

As an attempt to address the first problem, we utilise page counts by Google for estimating the probability of occurrences of the lexical units in W.

We consider the World Wide Web as a large general corpus and the Google search engine as a gateway for accessing the documents in the general corpus. Our choice of using Google to obtain the page count was merely motivated by its extensive coverage. In fact, it is possible to employ any search engines on the World Wide Web for this research. As for the second issue, we attempt to address the problem of determining the degree of collocational strength in terms of probabilities estimated using Google page count. We begin by defining the sample space, N as the set of all documents indexed by Google search engine. We can estimate the index size of Google, $|N|$ using function words as predictors. Function words such as *"a"*, *"is"* and *"with"*, as opposed to content words, appear with frequencies that are relatively stable over many different genres. Next, we perform random draws (i.e. trial) of documents from N. For each lexical unit $w \in W$, there will be a corresponding set of outcomes (i.e. events) from the draw. There will be three basic sets which are of interest to us:

Definition 2 *Basic events corresponding to each* $w \in W$*:*

- X *is the event that* a_x *occurs in the document*
- Y *is the event that* a_y *occurs in the document*
- S *is the event that* s *occurs in the document*

It should be obvious to the readers that since the documents in S have to contain all two units a_x and a_y, S is a subset of $X \cap Y$ or $S \subseteq X \cap Y$. It is worth noting that even though $S \subseteq X \cap Y$, it is highly unlikely that $S = X \cap Y$ since the two portions a_x and a_y may exist in the same document without being conjoined by b. Next, subscribing to the frequency interpretation of probability, we can obtain the probability of the events in Definition 2 in terms of Google page count:

$$P(X) = \frac{n_x}{|N|} \qquad (4)$$
$$P(Y) = \frac{n_y}{|N|}$$
$$P(S) = \frac{n_s}{|N|}$$

where n_x, n_y and n_s is the page count returned as the result of Google search using the term *[+"a_x"]*, *[+"a_y"]* and *[+"s"]*, respectively. The pair of quotes that encapsulates the search terms is the *phrase* operator, while the character "+" is the *required* operator supported by the Google search engine. As discussed earlier, the independence assumption required by certain information-theoretic measures and other Bayesian approaches may not always be valid, especially when we are dealing with linguistics. As such, $P(X \cap Y) \neq P(X)P(Y)$ since the occurrences of a_x and a_y in documents are inevitably governed by some hidden variables and hence, not independent. Following this, we define the probabilities for two new sets which result from applying some set operations on the basic events in Definition 2:

$$P(X \cap Y) = \frac{n_{xy}}{|N|} \qquad (5)$$
$$P(X \cap Y \setminus S) = P(X \cap Y) - P(S)$$

where n_{xy} is the page count returned by Google for the search using *[+"a_x" +"a_y"]*. Defining $P(X \cap Y)$ in terms of observable page counts, rather than a combination of two independent events will allow us to avoid any unnecessary assumption of independence.

Next, referring back to our main problem discussed in Definition 1, we are required to estimate the strength of collocation of the two units a_x and a_y. Since there is no standard metric for such measurement, we propose to address the problem from a probabilistic perspective. We introduce the probability that s is a stable lexical unit given the evidence s possesses:

Definition 3 *Probability of unithood:*

$$P(U|E) = \frac{P(E|U)P(U)}{P(E)}$$

where U is the event that s is a stable lexical unit and E is the evidences belonging to s. $P(U|E)$ is the posterior probability that s is a stable unit given the evidence E. $P(U)$ is the prior probability that s is a unit without any evidence, and $P(E)$ is the prior probability of evidences held by s. As we shall see later, these two prior probabilities will be immaterial in the final computation of unithood. Since s can either be a stable unit or not, we can state that,

$$P(\bar{U}|E) = 1 - P(U|E) \qquad (6)$$

where $\bar{U}$ is the event that s is not a stable lexical unit. Since $Odds = P/(1-P)$, we multiply both sides of Definition 3 by $(1 - P(U|E))^{-1}$ to obtain,

$$\frac{P(U|E)}{1 - P(U|E)} = \frac{P(E|U)P(U)}{P(E)(1 - P(U|E))} \quad (7)$$

By substituting Equation 6 in Equation 7 and later, applying the multiplication rule $P(\bar{U}|E)P(E) = P(E|\bar{U})P(\bar{U})$ to it, we will obtain:

$$\frac{P(U|E)}{P(\bar{U}|E)} = \frac{P(E|U)P(U)}{P(E|\bar{U})P(\bar{U})} \quad (8)$$

We proceed to take the log of the odds in Equation 8 (i.e. *logit*) to get:

$$\log \frac{P(E|U)}{P(E|\bar{U})} = \log \frac{P(U|E)}{P(\bar{U}|E)} - \log \frac{P(U)}{P(\bar{U})} \quad (9)$$

While it is obvious that certain words tend to co-occur more frequently than others (i.e. idioms and collocations), such phenomena are largely arbitrary (Smadja, 1993). This makes the task of deciding on what constitutes an acceptable collocation difficult. The only way to objectively identify stable lexical units is through observations in samples of the language (e.g. text corpus) (McKeown and Radev, 2000). In other words, assigning the apriori probability of collocational strength without empirical evidence is both subjective and difficult. As such, we are left with the option to assume that the probability of s being a stable unit and not being a stable unit without evidence is the same (i.e. $P(U) = P(\bar{U}) = 0.5$). As a result, the second term in Equation 9 evaluates to 0:

$$\log \frac{P(U|E)}{P(\bar{U}|E)} = \log \frac{P(E|U)}{P(E|\bar{U})} \quad (10)$$

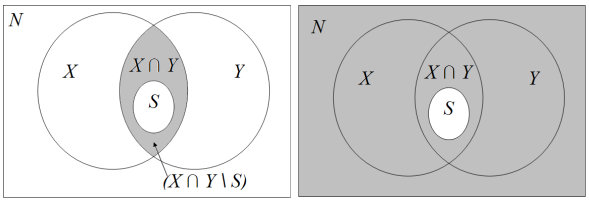
We introduce a new measure for determining the odds of s being a stable unit known as *Odds of Unithood (OU)*:

Definition 4 *Odds of unithood*

$$OU(s) = \log \frac{P(E|U)}{P(E|\bar{U})}$$

Assuming that the evidences in E are independent of one another, we can evaluate $OU(s)$ in terms of:

$$OU(s) = \log \frac{\prod_i P(e_i|U)}{\prod_i P(e_i|\bar{U})} = \sum_i \log \frac{P(e_i|U)}{P(e_i|\bar{U})} \quad (11)$$



(a) The area with darker shade is the set $X \cap Y \setminus S$. Computing the ratio of $P(S)$ and the probability of this area will give us the first evidence.

(b) The area with darker shade is the set S'. Computing the ratio of $P(S)$ and the probability of this area (i.e. $P(S') = 1 - P(S)$) will give us the second evidence.

Figure 1: The probability of the areas with darker shade are the denominators required by the evidences e_1 and e_2 for the estimation of $OU(s)$.

where e_i are individual evidences possessed by s.

With the introduction of Definition 4, we can examine the degree of collocational strength of a_x and a_y in forming s, mentioned in Definition 1 in terms of $OU(s)$. With the base of the log in Definition 4 more than 1, the upper and lower bound of $OU(s)$ would be $+\infty$ and $-\infty$, respectively. $OU(s) = +\infty$ and $OU(s) = -\infty$ corresponds to the highest and the lowest degree of stability of the two units a_x and a_y appearing as s, respectively. A high[1] $OU(s)$ would indicate the suitability for the two units a_x and a_y to be merged to form s. Ultimately, we have reduced the vague problem of the determination of unithood introduced in Definition 1 into a practical and computable solution in Definition 4. The evidences that we propose to employ for determining unithood are based on the occurrences of s, or the event S if the readers recall from Definition 2. We are interested in two types of occurrences of s, namely, the occurrence of s given that a_x and a_y have already occurred or $X \cap Y$, and the occurrence of s as it is in our sample space, N. We refer to the first evidence e_1 as *local occurrence*, while the second one e_2 as *global occurrence*. We will discuss the intuitive justification behind each type of occurrences. Each evidence e_i captures the occurrences of s within a different confinement. We will estimate these evidences in terms of the elementary probabilities already defined in Equations 4 and 5.

The first evidence e_1 captures the probability of occurrences of s within the confinement of a_x and a_y

[1] A subjective issue that may be determined using a threshold

or $X \cap Y$. As such, $P(e_1|U)$ can be interpreted as the probability of s occurring within $X \cap Y$ as a stable unit or $P(S|X \cap Y)$. On the other hand, $P(e_1|\bar{U})$ captures the probability of s occurring in $X \cap Y$ not as a unit. In other words, $P(e_1|\bar{U})$ is the probability of s not occurring in $X \cap Y$, or equivalently, equal to $P((X \cap Y \setminus S)|(X \cap Y))$. The set $X \cap Y \setminus S$ is shown as the area with darker shade in Figure 1(a). Let us define the odds based on the first evidence as:

$$O_L = \frac{P(e_1|U)}{P(e_1|\bar{U})} \tag{12}$$

Substituting $P(e_1|U) = P(S|X \cap Y)$ and $P(e_1|\bar{U}) = P((X \cap Y \setminus S)|(X \cap Y))$ into Equation 12 will give us:

$$\begin{aligned} O_L &= \frac{P(S|X \cap Y)}{P((X \cap Y \setminus S)|(X \cap Y))} \\ &= \frac{P(S \cap (X \cap Y))}{P(X \cap Y)} \frac{P(X \cap Y)}{P((X \cap Y \setminus S) \cap (X \cap Y))} \\ &= \frac{P(S \cap (X \cap Y))}{P((X \cap Y \setminus S) \cap (X \cap Y))} \end{aligned}$$

and since $S \subseteq (X \cap Y)$ and $(X \cap Y \setminus S) \subseteq (X \cap Y)$,

$$O_L = \frac{P(S)}{P(X \cap Y \setminus S)} \quad if(P(X \cap Y \setminus S) \neq 0)$$

and $O_L = 1$ if $P(X \cap Y \setminus S) = 0$.

The second evidence e_2 captures the probability of occurrences of s without confinement. If s is a stable unit, then its probability of occurrence in the sample space would simply be $P(S)$. On the other hand, if s occurs not as a unit, then its probability of non-occurrence is $1 - P(S)$. The complement of S, which is the set S' is shown as the area with darker shade in Figure 1(b). Let us define the odds based on the second evidence as:

$$O_G = \frac{P(e_2|U)}{P(e_2|\bar{U})} \tag{13}$$

Substituting $P(e_2|U) = P(S)$ and $P(e_2|\bar{U}) = 1 - P(S)$ into Equation 13 will give us:

$$O_G = \frac{P(S)}{1 - P(S)}$$

Intuitively, the first evidence attempts to capture the extent to which the existence of the two lexical units a_x and a_y is attributable to s. Referring back to O_L, whenever the denominator $P(X \cap Y \setminus S)$ becomes less than $P(S)$, we can deduce that a_x and a_y actually exist together as s more than in other forms. At one extreme when $P(X \cap Y \setminus S) = 0$, we can conclude that the co-occurrence of a_x and a_y is exclusively for s. As such, we can also refer to O_L as a measure of exclusivity for the use of a_x and a_y with respect to s. This first evidence is a good indication for the unithood of s since the more the existence of a_x and a_y is attributed to s, the stronger the collocational strength of s becomes. Concerning the second evidence, O_G attempts to capture the extent to which s occurs in general usage (i.e. World Wide Web). We can consider O_G as a measure of pervasiveness for the use of s. As s becomes more widely used in text, the numerator in O_G will increase. This provides a good indication on the unithood of s since the more s appears in usage, the likelier it becomes that s is a stable unit instead of an occurrence by chance when a_x and a_y are located next to each other. As a result, the derivation of $OU(s)$ using O_L and O_G will ensure a comprehensive way of determining unithood.

Finally, expanding $OU(s)$ in Equation 11 using Equations 12 and 13 will give us:

$$\begin{aligned} OU(s) &= \log O_L + \log O_G \\ &= \log \frac{P(S)}{P(X \cap Y \setminus S)} + \log \frac{P(S)}{1 - P(S)} \end{aligned} \tag{14}$$

As such, the decision on whether a_x and a_y should be merged to form s can be made based solely on the *Odds of Unithood (OU)* defined in Equation 14. We will merge a_x and a_y if their odds of unithood exceeds a certain threshold, OU_T.

4 Evaluations and Discussions

For this evaluation, we employed 500 news articles from Reuters in the health domain gathered between December 2006 to May 2007. These 500 articles are fed into the Stanford Parser whose output is then used by our head-driven left-right filter (Wong, 2005; Wong et al., 2007b) to extract word sequences in the form of nouns and noun phrases. Pairs of word sequences (i.e. a_x and a_y) located immediately next to each other, or separated by a preposition or the conjunction *"and"* in the same sentence are mea-

sured for their unithood. Using the 500 news articles, we managed to obtain 1,825 pairs of words to be tested for unithood.

We performed a comparative study of our new probabilistic approach against the empirically-derived unithood function described in Equation 2. Two experiments were conducted. In the first one, we assessed our probabilistically-derived measure $OU(s)$ as described in Equation 14 where the decisions on whether or not to merge the 1,825 pairs are done automatically. These decisions are known as the *actual results*. At the same time, we inspected the same list manually to decide on the merging of all the pairs. These decisions are known as the *ideal results*. The threshold OU_T employed for our evaluation is determined empirically through experiments and is set to -8.39. However, since only one threshold is involved in deciding mergeability, training algorithms and data sets may be employed to automatically decide on an optimal number. This option is beyond the scope of this paper. The actual and ideal results for this first experiment are organised into a contingency table (not shown here) for identifying the true and the false positives, and the true and the false negatives. In the second experiment, we conducted the same assessment as carried out in the first one but the decisions to merge the 1,825 pairs are based on the $UH(a_x, a_y)$ function described in Equation 2. The thresholds required for this function are based on the values suggested by (Wong et al., 2007b), namely, $MI^+ = 0.9$, $MI^- = 0.02$, $ID_T = 6$, $IDR^+ = 1.35$, and $IDR^- = 0.93$.

Table 1: The performance of $OU(s)$ (from Experiment 1) and $UH(a_x, a_y)$ (from Experiment 2) in terms of precision, recall and accuracy. The last column shows the difference in the performance of Experiment 1 and 2.

	Experiment 1 using *OU(s)*	Experiment 2 using $UH(a_x,a_y)$	Difference (Experiment 1-Experiment 2)
Precision	100.00%	97.37%	2.63%
Recall	95.83%	92.50%	3.33%
Accuracy	97.26%	94.52%	2.74%

Using the results from the contingency tables, we computed the precision, recall and accuracy for the two measures under evaluation. Table 1 summarises the performance of $OU(s)$ and $UH(a_x, a_y)$ in determining the unithood of 1,825 pairs of lexical units. One will notice that our new measure $OU(s)$ outperformed the empirically-derived function $UH(a_x, a_y)$ in all aspects, with an improvement of 2.63%, 3.33% and 2.74% for precision, recall and accuracy, respectively. Our new measure achieved a 100% precision with a lower recall at 95.83%. As with any measures that employ thresholds as a cut-off point in accepting or rejecting certain decisions, we can improve the recall of $OU(s)$ by decreasing the threshold OU_T. In this way, there will be less false negatives (i.e. pairs which are supposed to be merged but are not) and hence, increases the recall rate. Unfortunately, recall will improve at the expense of precision since the number of false positives will definitely increase from the existing 0. Since our application (i.e. ontology learning) requires perfect precision in determining the unithood of word sequences, $OU(s)$ is the ideal candidate. Moreover, with only one threshold (i.e. OU_T) required in controlling the function of $OU(s)$, we are able to reduce the amount of time and effort spent on optimising our results.

5 Conclusion and Future Work

In this paper, we highlighted the significance of unithood and that its measurement should be given equal attention by researchers in term extraction. We focused on the development of a new approach that is independent of influences of termhood measurement. We proposed a new probabilistically-derived measure which provide a dedicated way to determine the unithood of word sequences. We refer to this measure as the *Odds of Unithood (OU)*. OU is derived using Bayes Theorem and is founded upon two evidences, namely, *local occurrence* and *global occurrence*. Elementary probabilities estimated using page counts from the Google search engine are utilised to quantify the two evidences. The new probabilistically-derived measure OU is then evaluated against an existing empirical function known as *Unithood (UH)*. Our new measure OU achieved a precision and a recall of 100% and 95.83% respectively, with an accuracy at 97.26% in measuring the unithood of 1,825 test cases. OU outperformed UH by 2.63%, 3.33% and 2.74% in terms of precision,

recall and accuracy, respectively. Moreover, our new measure requires only one threshold, as compared to five in UH to control the mergeability decision.

More work is required to establish the *coverage* and the *depth* of the World Wide Web with regards to the determination of unithood. While the Web has demonstrated reasonable strength in handling general news articles, we have yet to study its appropriateness in dealing with unithood determination for technical text (i.e. the depth of the Web). Similarly, it remains a question the extent to which the Web is able to satisfy the requirement of unithood determination for a wider range of genres (i.e. the coverage of the Web). Studies on the effect of noises (e.g. keyword spamming) and multiple word senses on unithood determination using the Web is another future research direction.

Acknowledgement

This research was supported by the Australian Endeavour International Postgraduate Research Scholarship, and the Research Grant 2006 by the University of Western Australia.

References

K. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

T. Dunning. 1994. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

K. Frantzi. 1997. Incorporating context information for the extraction of terms. In *Proceedings of the 35th Annual Meeting on Association for Computational Linguistics*, Spain.

A. Franz. 1997. Independence assumptions considered harmful. In *Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics*, Madrid, Spain.

K. Kageura and B. Umino. 1996. Methods of automatic term recognition: A review. *Terminology*, 3(2):259–289.

F. Keller, M. Lapata, and O. Ourioupina. 2002. Using the web to overcome data sparseness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia.

C. Kit. 2002. Corpus tools for retrieving and deriving termhood evidence. In *Proceedings of the 5th East Asia Forum of Terminology*, Haikou, China.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

D. Kurz and F. Xu. 2002. Text mining for the extraction of domain relevant terms and term collocations. In *Proceedings of the International Workshop on Computational Approaches to Collocations*, Vienna.

K. McKeown and D. Radev. 2000. Collocations. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*. Marcel Dekker.

H. Nakagawa and T. Mori. 2002. A simple but powerful automatic term extraction method. In *Proceedings of the International Conference On Computational Linguistics (COLING)*.

V. Seretan, L. Nerima, and E. Wehrli. 2004. Using the web as a corpus for the syntactic-based collocation identification. In *Proceedings of the International Conference on on Language Resources and Evaluation (LREC)*, Lisbon, Portugal.

F. Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.

W. Wong, W. Liu, and M. Bennamoun. 2007a. Determining termhood for learning domain ontologies in a probabilistic framework. In *Proceedings of the 6th Australasian Conference on Data Mining (AusDM)*, Gold Coast.

W. Wong, W. Liu, and M. Bennamoun. 2007b. Determining the unithood of word sequences using mutual information and independence measure. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING)*, Melbourne, Australia.

W. Wong. 2005. Practical approach to knowledge-based question answering with natural language understanding and advanced reasoning. Master's thesis, National Technical University College of Malaysia, arXiv:cs.CL/0707.3559.

Lexical Chains as Document Features

Dinakar Jayarajan, Dipti Deodhare
Centre for Artificial Intelligence and Robotics,
Defence R & D Organisation,
Bangalore, INDIA.
[dinakarj, dipti]@cair.drdo.in

B Ravindran
Dept. of CSE,
IIT Madras,
Chennai, INDIA.
ravi@cse.iitm.ac.in

Abstract

Document clustering and classification is usually done by representing the documents using a bag of words scheme. This scheme ignores many of the linguistic and semantic features contained in text documents. We propose here an alternative representation for documents using Lexical Chains. We compare the performance of the new representation against the old one on a clustering task. We show that Lexical Chain based features give better results than the Bag of Words based features, while achieving almost 30% reduction in the dimensionality of the feature vectors resulting in faster execution of the algorithms.

1 Introduction

Text data usually contains complex semantic information which is communicated using a combination of words. Ideally, the representation used should capture and reflect this fact in order to semantically drive the clustering algorithm and obtain better results.

The Bag of Words (BoW) (Salton et al., 1975) scheme is a very popular scheme which has been used for representing documents. But, this scheme ignores many of the linguistic and semantic features contained in text documents. This paper explores an alternative representation for documents, using *lexical chains*, which encodes some of the semantic information contained in the document. This representation results in improved performance on the clustering tasks and achieves a drastic reduction in the size of the feature space as well.

The BoW scheme was originally designed for the Information Retrieval domain (Salton, 1989) where the aim was to 'index' the document and not necessarily to model the topic distribution. This representation has since been adopted as the *defacto* document representation scheme for supervised and unsupervised learning on documents. The BoW scheme represents features as an unordered set of words contained in the document, along with their frequency count.

The BoW scheme assumes that the distribution of words in a document reflect the underlying distribution of topics and hence if the documents are grouped on the basis of the similarity of the words contained in them, it will implicitly result in a clustering based on topics. This representation, using a simple frequency count alone, does not capture all the underlying information present in the documents. Moreover, it ignores information such as position, relations and co-occurrences among the words. In addition, the feature space formed will be very huge and sparse resulting in time and space costs as well.

Lexical Chaining is a technique which seeks to identify and exploit the semantic relatedness of words in a document. It is based on the phenomenon of *lexical cohesion* (Halliday and Hasan, 1976) and works on the premise that semantically related words co-occur close together in a passage more than "just by chance". Lexical chaining is the process of identifying and grouping such words together to form chains which in turn will help in identifying and representing the topic and content of the document.

Lexical chains have been used as an intermediate representation of text for various tasks such as au-

tomatic text summarisation (Barzilay and Elhadad, 1997; Silber and McCoy, 2002), malapropism detection and correction (Hirst and St-Onge, 1997), and hypertext construction (Green, 1998). An algorithm for computing lexical chains was first given by (Morris and Hirst, 1991) using the Roget's Thesaurus (Kirkpatrick, 1998). Since an electronic version of the Roget's Thesaurus was not available then, later algorithms were based on the WordNet lexical database (Fellbaum, 1998).

We present here a two pass algorithm to compute a representation of documents using lexical chains and use these lexical chains to derive feature vectors. These lexical chain based feature vectors are used to cluster the documents using two different algorithms - *k*-Means and Co-clustering. *k*-Means is a well studied clustering algorithm widely used in the text domain. Co-clustering, also known as bi-clustering (Madeira and Oliveira, 2004), is a clustering approach which was developed in the bioinformatics domain for clustering gene expressions. Since the text domain shares a lot of characteristics (high dimensionality, sparsity, *etc.*) of gene expression data, a lot of interest has been generated recently in applying the co-clustering approaches (Dhillon et al., 2003) to the text domain with promising results. Co-clustering (Dhillon et al., 2003; Sra et al., 2004) exploits the duality between rows and columns of the document-term matrix used to represent the features, by simultaneously clustering both the rows and columns.

We compare the clustering results obtained from document features extracted using lexical chains against those obtained by using the traditional method of bag of words.

2 Lexical Chains

Lexical chains are groups of words which exhibit lexical cohesion. Cohesion as given by (Halliday and Hasan, 1976) is a way of getting text to "hang together as a whole". Lexical cohesion is exhibited through cohesive relations. They (Halliday and Hasan, 1976) have classified these relations as:

1. Reiteration with identity of reference
2. Reiteration without identity of reference
3. Reiteration by means of super ordinate
4. Systematic semantic relation
5. Non systematic semantic relation

The first three relations involve reiteration which includes repetition of the same word in the same sense (*e.g.*, car and car), the use of a synonym for a word (*e.g.*, car and automobile) and the use of hypernyms (or hyponyms) for a word (*e.g.*, car and vehicle) respectively. The last two relations involve collocations *i.e*, semantic relationships between words that often co-occur (*e.g.*, football and foul). Lexical chains in a text are identified by the presence of strong semantic relations between the words in the text.

Algorithms for building lexical chains work by considering candidate words for inclusion in the chains constructed so far. Usually these candidate words are nouns and compound nouns. Lexical Chains can be computed at various granularities - across sentences, paragraphs or documents. In general, to compute lexical chains, each candidate word in the sentence/paragraph/document is compared, with each lexical chain identified so far. If a candidate word has a 'cohesive relation' with the words in the chain it is added to the chain. On the other hand, if a candidate word is not related to any of the chains, a new chain is created for the candidate word. Thus a lexical chain is made up of a set of semantically related words. The lexical chains obtained are then evaluated based on a suitable criteria and the better chains are selected and used to further processing. Naturally, the computation of lexical chains is predicated on the availability of a suitable database which maps relations between words.

Several algorithms have been proposed for computing lexical chains. Prominent among them are those by (Hirst and St-Onge, 1997; Barzilay and Elhadad, 1997; Silber and McCoy, 2002; Jarmasz and Szpakowicz, 2003). Except for the one by Jarmasz and Szpakowicz, all others use WordNet (Fellbaum, 1998) to identify relations among words. A brief overview of these algorithms is given in (Jayarajan et al., 2007).

WordNet is a lexical database which organises words into synonym sets or *synsets*. Each synset contains one or more words that have the same meaning. A word may appear in many synsets, depending on the number of senses that it has. The

synsets are connected by links that indicate different semantic relations such as generalisation (hypernyms), specialisation (hyponyms), part relations (holonyms[1] and meronyms[2]), *etc*.

Our approach to computing lexical chains differs from those listed above and is described in the next section.

3 Lexical Chains based Feature Vectors

All the algorithms mentioned in the previous section, try to disambiguate the sense of the word as part of the chaining process. Both Word Sense Disambiguation (WSD) and lexical chaining are very profound processes. The aim of computing the lexical chains here is to try and identify the topics in a document. If WSD has be performed as an implicit step in the lexical chain computing algorithm, it tends to deteriorate the outcome of both. We feel that the words should be disambiguated by looking at their context in a sentence/paragraph as a whole. As such, we propose to perform WSD as a preprocessing step, before the word is considered for lexical chaining. We use an algorithm by (Patwardhan et al., 2003) to disambiguate the senses of the words in reference to Wordnet. We then filter out all non-noun words identified in the WSD stage. This is based on the assumption that nouns are better at reflecting the topics contained in a document than the other parts of speech. The result is a set of nouns which appear in the text along with its sense. We refer to these as 'candidate words'.

Our algorithm is based on the WordNet Lexical Database. WordNet is used to identify the relations among the words. We use only the identity and synonymy relations to compute the chains. A word has a identity or synonymy relation with another word, only if both the words occur in the same synset in Wordnet. Empirically, we found that usage of only these two relations, resulted in chains representing crisp topics.

A lexical chain contains a list of words which are related to each other and is identified using a unique numeric identifier. Each word in turn is represented as a 4-tuple <term, pos, sense, rel>, where 'pos' is the part-of-speech of the term, 'sense' is the Wordnet sense number and 'rel' is the relation of this word to the chain. In this case, we treat the two relations - identity and synonymy, as a single relation and hence this is uniformly 'IS' for all the words.

Definition 1 *Length of a lexical chain L is defined as the number of words in the chain.*

$$length(L) = \text{Number of Words in Chain L} \quad (1)$$

The length of a lexical chain is an indicator of the strength of the chain in representing a topic. Dominant topics/information will have long chains, while stray information will form extremely short chains. Each occurrence of a word in a document, will increase the length of the chain by one. Thus, the length of a chain gives a composite measure of the number of documents in which the chain occurs and the number of occurences of words in the chain in these documents.

3.1 Feature Vector Computation

We use a two pass algorithm to generate feature vectors based on lexical chains. Our algorithm works by maintaining a global set of lexical chains, each of which represents a topic. Initially, the global list is empty. In the first pass we identify all possible lexical chains for that document. This is achieved by comparing the candidate words of each document with the global list to identify those chains with which it has a identity or synonymy relation. If no chains are identified, a new chain is created and put in the global list. The candidate word is then added to the chain. At the end of this pass, we obtain a global set which lists all the chains contained in all the documents. The algorithm is presented in Algorithm 1.

In the second pass we select a subset of chains from the global set, which can be used to represent the document. We define and use a measure to evaluate and select the chains as follows:

Definition 2 *The significance of a lexical chain L in a Global set G is defined as*

[1]part of, member of, substance of relations, *e.g.,* 'wheel' is part of a 'vehicle'

[2]has part, has member, has substance relations, *e.g.,* 'wheel' has part 'rim'

Algorithm 1 Identify Chains

```
1: Maintain a global set of lexical chains, initialised to a Null set
2: for each document do
3:    for each candidate word in document do
4:       Identify lexical chains in global set with which the word has a identity/synonym relation
5:       if No chain is identified then
6:          Create a new chain for this word and insert in global set
7:       end if
8:       Add word to the identified/created chains in Global Set
9:    end for
10: end for
```

Algorithm 2 Select Chains and Generate FV

```
   for each document do
2:    Initialise feature vector to zero
      for each candidate word in document do
4:       Identify lexical chains in global set with which the word has a identity/synonym relation
      end for
6:    Compute threshold for document
      for each identified chain in global set do
8:       if utility of chain greater than threshold then
            Set component corresponding to chain in feature vector to 1
10:      end if
      end for
12: end for
```

$$sig(L) = -\frac{length(L)}{\sum\limits_{l\ \epsilon\ G} length(l)} . \log_2 \frac{length(L)}{\sum\limits_{l\ \epsilon\ G} length(l)} \tag{2}$$

The significance of a chain L measures how randomly the chain appears in the global set G. This measure helps in identifying good chains from weak, random ones in the global set. In effect, $sig(L)$ will select those chains which are not abnormally long or short with respect to the distributions of chains in the global set.

Definition 3 *A candidate word W is related to a lexical chain L if W has an identity or synonym relation with L.*

$$\begin{aligned} related(W, L) &= 1,\ W\ and\ L\ are\ related \\ &= 0,\ otherwise \end{aligned} \tag{3}$$

Definition 4 *The utility of a lexical chain L to a document D is defined as*

$$util(L, D) = sig(L) . \sum_{all\ w\ \epsilon\ D} related(w, L) \tag{4}$$

The utility of a chain L is a measure of how good L will be in representing the document. This is based on the observation that long chains are better than short ones. This measure will prefer 'good' chains from the global set, which are related to a large number of candidate words in the document.

We select and assign to the document all those chains which cross a threshold on the utility of the chain. Empirically, we found that using a threshold of 'half the average' utility for a document gave good results. For a document D, let the set of all lexical chains assignable to D be $G' \subset$ GlobalSet G. The threshold for D is computed as

$$threshold(D) = \frac{\sum\limits_{l\ \epsilon\ G'} util(l, D)}{2\ .\ |G'|} \tag{5}$$

The lexical chains in the global list form the components of the feature vectors. We use a binary valued scheme, where in we put a 1 corresponding to a chain if the chain is assigned to the document and 0 otherwise. Essentially, what we obtain here is a feature vector of size equal to the number of lexical chains in the global list. The second pass of the algorithm is listed in Algorithm 2.

Cluster	Document Ids
college atheists	53675, 53357, 53540
amusing atheists and anarchists	53402, 53351
islam & dress code for women	51212, 51216, 51318

Table 2: Example of the classes obtained from grouping the documents using the subject line

4 Experiments

We use the 20 Newsgroups (Rennie, 1995) dataset to evaluate the utility of representing documents using the lexical chains (*lexchains*) scheme. The 20 Newsgroups (20NG) corpus is a collection of usenet messages spread across 20 Usenet groups. These messages are written by a wide population of net users and represent a good variation in writing styles, choice of words and grammar. Thus, we feel that the 20NG is a representative corpus for the purpose. We derive three datasets from three distinct groups of the 20NG corpus - *comp.windows.x* (`cwx`), *rec.motorcycles* (`rm`) and *alt.atheism* (`aa`). The statistics of the datasets is given in Table 1.

The documents in each dataset are further grouped on the basis of their subject lines. This grouping into classes is used as the gold standard for evaluating the clustering algorithms. An example of the groups formed for the `aa` dataset is shown in Table 2.

We prepared the dataset for feature extraction by removing the complete header including the subject line and used only the body portion of the messages to compute the features. We extracted features on this cleaned data using both the BoW and *lexchains* scheme. For the BoW scheme, we first tokenised the document, filtered out the stopwords using the list obtained from (Fox, 1992) and further stemmed them using a Porter Stemmer (Porter, 1980). The feature vectors were then computed using the *tf.idf* scheme. We refer to the feature vectors thus obtained as `cwx-BoW`, `rm-BoW` and `aa-BoW` for the `cwx`, `rm` and `aa` datasets respectively. *Lexchains* based features were derived as described in Section 3.1 and are analogously referred to here as `cwx-lc`, `rm-lc` and `aa-lc`. This results in a total of six datasets. The dimensions of the feature vectors obtained are summarised in Table 3. It can be noted that the size of the feature vectors are reduced by more than 30% with the *lexchains* based features.

These six datasets were clustered using the *k*-Means and Co-clustering algorithms. The *k*-Means implementation in Matlab was used and *k* was set to 649, 340 and 196 for `cwx`, `rm` and `aa` respectively and reflects the number of classes identified in the gold standard (*ref.* Table. 1). The co-clustering experiments were done using the Minimum Sum-Squared Residue Co-clustering algorithm (Sra et al., 2004) with the number of row clusters set to the same values as given to the *k*-Means algorithm.

We use a normalised edit distance based measure to evaluate the goodness of the clusters. This measure is a variant of the one used by (Pantel and Lin, 2002), which defines an edit distance as the number of merge, move and copy operations required to transform the resulting clusters to the gold standard. Initially, if there are c classes in the gold standard, we create c empty clusters. The measure then merges each resulting cluster to the cluster in the gold standard with which it has maximum overlap, breaking ties randomly. Thus, the merge operation attempts to bring the obtained clusters as close as possible to the gold standard as a whole. Subsequently, the move and copy operations are used to

Collection	# Classes	# Documents
`comp.windows.x`	649	980
`alt.atheism`	196	799
`rec.motorcycles`	340	994

Table 1: Dataset Statistics

	BoW	*lexchain*	Reduction
`cwx`	12767	4569	64%
`aa`	8881	5980	32%
`rm`	8675	5288	39%

Table 3: Dimensionality of the Feature Vectors

	k-Means	Co-cluster	Time (secs)
cwx-BoW	203 (0.21)	**140 (0.14)**	1529
cwx-lc	**179 (0.18)**	158 (0.16)	**201**
aa-BoW	85 (0.11)	110 (0.13)	869
aa-lc	**60 (0.08)**	**82 (0.10)**	**221**
rm-BoW	**113 (0.11)**	208 (0.21)	1177
rm-lc	127 (0.12)	**144 (0.14)**	**229**

Table 4: Edit distance between obtained clusters and gold standard. Normalised edit distances are given in parenthesis. The fourth column gives runtime for the co-clustering algorithm, averaged over four runs. (*For all cases, lower is better.*)

move (copy) the documents around so that they finally match the gold standard.

We observed that the merge operation would inevitably add as many clusters as there are in the gold standard to the final count, skewing the results. Hence, we define the edit distance as only the number of move and copy[3] operations required to convert the obtained clusters to that of the gold standard. In effect, it measures the number of documents which are misplaced with respect to the gold standard. The obtained edit distance is normalised by dividing it with the number of documents in the dataset. This will normalise the value of the measure to range between 0 and 1. The lower the value of this measure, the closer the obtained clustering is to the gold standard.

The results are enumerated in Table 4. The *lexchains* based document feature gives an improvement of upto 33% over the BoW representation while achieving a reduction in dimensions of the feature vectors by more than 30% (*ref.* Table 3). We performed run time studies on the dataset using the co-clustering algorithm. The runtimes are averaged over four runs. It can be seen that a speedup of more than 74% is achieved with the *lexchain* based features[4].

Thus, the results show that the running time of the clustering algorithms is drastically reduced while maintaining or improving the clustering performance through the use of *lexchain* based features.

[3]The copy count will be included only in the case of overlapping clusters, which happens if a document is in more than one cluster.

[4]It was observed empirically that the time required to compute both the BoW and *lexchain* features are nearly the same and hence can be ignored.

4.1 Discussion

A document is not just a bunch of loose words. Each word in a document contributes to some aspect of the overall semantics of the document. Classification and clustering algorithms seek to group the documents based on its semantics. The BoW scheme inherently throws away a lot of information, which would have otherwise been useful in discerning the semantics of the document. The BoW representation fails to capture and represent these semantics resulting in a less accurate representation for the documents. This fact is reflected by higher edit distance in the case of BoW based clustering in Table 4.

Earlier, Hatzivassiloglou, *et. al.* (Hatzivassiloglou et al., 2000) had studied the effects of linguistically motivated features on clustering algorithms. They had explored two linguistically motivated features - noun phrase heads and proper names and compared these against the bag of words representation. They had reported that the BoW representation was better than linguistically motivated features. We believe that noun phrase heads and proper names are inadequate representations of the semantics of a document and a more composite representation is required to obtain better results on semantically oriented tasks.

Lexical chains appear to be capable of doing this to a certain extent. During the process of computing and selecting the lexical chains, we are implicitly trying to decode the semantics of the documents. Lexical chains work on the basic premise that a document describes topics through a combination of words and these words will exhibit a cohesion among them. This cohesion can be identified using a resource such as WordNet. In the process, lexical chains capture some amount of the semantics contained in the documents, resulting in a better performance in subsequent processing of the documents.

5 Conclusion

We have shown that semantically motived features, such as lexical chains, provide a better representation for the documents, resulting in comparable or

better performance on clustering tasks while effecting a drastic reduction in time and space complexity.

Even though the lexical chains manage to represent the semantics to a certain extent, we feel it can be further enhanced by more involved processing. A comparision of lexical chains based representation with other document representation schemes such as LSA also warrants investigation.

6 Acknowledgement

The authors would like to thank Director, CAIR for the encouragement and support given for this work. The authors would also like to thank the three anonymous reviewers, whose detailed comments helped improve this work.

References

Regina Barzilay and M. Elhadad. 1997. Using lexical chains for text summarization. In *In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, Spain.*

Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. 2003. Information-theoretic co-clustering. In *Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2003)*, pages 89–98.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Christopher Fox, 1992. *Information retrieval: data structures and algorithms*, chapter Lexical Analysis and Stoplists. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Stephen J Green. 1998. Automatically generating hypertext in newspaper articles by computing semantic relatedness. In D.M.W. Powers, editor, *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language*.

M. A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman.

Vasileios Hatzivassiloglou, Luis Gravano, and Ankineedu Maganti. 2000. An investigation of linguistic features and clustering algorithms for topical document clustering. In *SIGIR 2000*, pages 224–231.

Graeme Hirst and David St-Onge. 1997. Lexical chains as representation of context for the detection and correction of malapropisms. In C. Fellbaum, editor, *WordNet:An electronic lexical database and some of its applications.* The MIT Press, Cambrige, MA.

Mario Jarmasz and Stan Szpakowicz. 2003. Not as easy as it seems: Automating the construction of lexical chains using roget's thesaurus. In *Proceedings of the 16th Canadian Conference on Artificial Intelligence*.

Dinakar Jayarajan, Dipti Deodhare, B Ravindran, and Sandipan Sarkar. 2007. Document clustering using lexical chains. In *Proceedings of the Workshop on Text Mining & Link Analysis (TextLink 2007), Hyderbad, INDIA*, January.

B Kirkpatrick. 1998. *Roget's Thesaurus of English Words and Phrases*. Penguin.

Sara C. Madeira and Arlindo L. Oliveira. 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Patrick Pantel and Dekang Lin. 2002. Document clustering with committees. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 199–206, New York, NY, USA. ACM Press.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics (CiCLING-03), Mexico City, Mexico.*

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*.

Jason Rennie. 1995. 20 Newsgroups dataset. Online: http://people.csail.mit.edu/jrennie/20Newsgroups/.

Gerard Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Gerard Salton. 1989. *Automatic Text Processing – The Transformation, Analysis, and Retrieval of Information by Computer*. Addison–Wesley.

H. Gregory Silber and Kathleen F. McCoy. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496.

Suvrit Sra, Hyuk Cho, Inderjit S. Dhillon, and Yuqiang Guan. 2004. Minimum sum-squared residue coclustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA, April 22-24, 2004*.

Entity-driven Rewrite for Multi-document Summarization

Ani Nenkova
University of Pennsylvania
Department of Computer and Information Science
nenkova@seas.upenn.edu

Abstract

In this paper we explore the benefits from and shortcomings of entity-driven noun phrase rewriting for multi-document summarization of news. The approach leads to 20% to 50% different content in the summary in comparison to an extractive summary produced using the same underlying approach, showing the promise the technique has to offer. In addition, summaries produced using entity-driven rewrite have higher linguistic quality than a comparison non-extractive system. Some improvement is also seen in content selection over extractive summarization as measured by pyramid method evaluation.

1 Introduction

Two of the key components of effective summarizations are the ability to identify important points in the text and to adequately reword the original text in order to convey these points. Automatic text summarization approaches have offered reasonably well-performing approximations for identifiying important sentences (Lin and Hovy, 2002; Schiffman et al., 2002; Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Daumé III and Marcu, 2006) but, not surprisingly, text (re)generation has been a major challange despite some work on sub-sentential modification (Jing and McKeown, 2000; Knight and Marcu, 2000; Barzilay and McKeown, 2005). An additional drawback of extractive approaches is that estimates for the importance of larger text units such as sentences depend on the length of the sentence (Nenkova et al., 2006).

Sentence simplification or compaction algorithms are driven mainly by grammaticality considerations. Whether approaches for estimating importance can be applied to units smaller than sentences and used in text rewrite in the summary production is a question that remains unanswered. The option to operate on smaller units, which can be mixed and matched from the input to give novel combinations in the summary, offers several possible advantages.

Improve content Sometimes sentences in the input can contain both information that is very appropriate to include in a summary and information that should not appear in a summary. Being able to remove unnecessary parts can free up space for better content. Similarly, a sentence might be good overall, but could be further improved if more details about an entity or event are added in. Overall, a summarizer capable of operating on subsentential units would in principle be better at content selection.

Improve readability Linguistic quality evaluation of automatic summaries in the Document Understanding Conference reveals that summarizers perform rather poorly on several readability aspects, including referential clarity. The gap between human and automatic performance is much larger for linguistic quality aspects than for content selection. In more than half of the automatic summaries there were entities for which it was not clear what/who they were and how they were related to the story. The ability to add in descriptions for entities in the summaries could improve the referential clarity of summaries and can be achieved through text rewrite

of subsentential units.

IP issues Another very practical reason to be interested in altering the original wording of sentences in summaries in a news browsing system involves intellectual property issues. Newspapers are not willing to allow verbatim usage of long passages of their articles on commercial websites. Being able to change the original wording can thus allow companies to include longer than one sentence summaries, which would increase user satisfaction (McKeown et al., 2005).

These considerations serve as direct motivation for exploring how a simple but effective summarizer framework can accommodate noun phrase rewrite in multi-document summarization of news. The idea is for each sentence in a summary to automatically examine the noun phrases in it and decide if a different noun phrase is more informative and should be included in the sentence in place of the original. Consider the following example:

Sentence 1 *The arrest* caused an international controversy.

Sentence 2 *The arrest in London of former Chilean dictator Augusto Pinochet* caused an international controversy.

Now, consider the situation where we need to express in a summary that the arrest was controversial and this is the first sentence in the summary, and sentence 1 is available in the input ("The arrest caused an international controversy"), as well as an unrelated sentence such as "The arrest in London of former Chilean dictator Augusto Pinochet was widely discussed in the British press". NP rewrite can allow us to form the rewritten sentence 2, which would be a much more informative first sentence for the summary: "The arrest in London of former Chilean dictator Augusto Pinochet caused an international controversy". Similarly, if sentence 2 is available in the input and it is selected in the summary after a sentence that expresses the fact that the arrest took place, it will be more appropriate to rewrite sentence 2 into sentence 1 for inclusion in the summary.

This example shows the potential power of noun phrase rewrite. It also suggests that context will play a role in the rewrite process, since different noun phrase realizations will be most appropriate depending on what has been said in the summary up to the point at which rewrite takes place.

2 NP-rewrite enhanced frequency summarizer

Frequency and frequency-related measures of importance have been traditionally used in text summarization as indicators of importance (Luhn, 1958; Lin and Hovy, 2000; Conroy et al., 2006). Notably, a greedy frequency-driven approach leads to very good results in content selection (Nenkova et al., 2006). In this approach sentence importance is measured as a function of the frequency in the input of the content words in that sentence. The most important sentence is selected, the weight of words in it are adjusted, and sentence weights are recomputed for the new weights beofre selecting the next sentence.

This conceptually simple summarization approach can readily be extended to include NP rewrite and allow us to examine the effect of rewrite capabilities on overall content selection and readability. The specific algorithm for frequency-driven summarization and rewrite is as follows:

Step 1 Estimate the importance of each content word w_i based on its frequency in the input n_i, $p(w_i) = \frac{n_i}{N}$.

Step 2 For each sentence S_j in the input, estimate its importance based on the words in the sentence $w_i \in S_j$: the weight of the sentence is equal to the average weight of content words appearing in it.

$$Weight(S_j) = \frac{\sum_{w_i \in S_j} p(w_i)}{|w_i \in S_j|}$$

Step 3 Select the sentence with the highest weight.

Step 4 For each maximum noun phrase NP_k in the selected sentence

4.1 For each coreferring noun phrase NP_i, such that $NP_i \equiv NP_k$ from all input documents, compute a weight $Weight(NP_i) = F_{RW}(w_r \in NP_i)$.

4.2 Select the noun phrase with the highest weight and insert it in the sentence in

place of the original NP. In case of ties, select the shorter noun phrase.

Step 5 For each content word in the rewritten sentence, update its weight by setting it to 0.

Step 6 If the desired summary length has not been reached, go to step 2.

Step 4 is the NP rewriting step. The function F_{RW} is the rewrite composition function that assigns weights to noun phrases based on the importance of words that appear in the noun phrase. The two options that we explore here are $F_{RW} \equiv Avr$ and $F_{RW} \equiv Sum$; the weight of an NP equals the average weight or sum of weights of content words in the NP respectively. The two selections lead to different behavior in rewrite. $F_{RW} \equiv Avr$ will generally prefer the shorter noun phrases, typically consisting of just the noun phrase head and it will overall tend to reduce the selected sentence. $F_{RW} \equiv Sum$ will behave quite differently: it will insert relevant information that has not been conveyed by the summary so far (add a longer noun phrase) and will reduce the NP if the words in it already appear in the summary. This means that $F_{RW} \equiv Sum$ will have the behavior close to what we expect for entity-centric rewrite: inluding more descriptive information at the first mention of the entity, and using shorter references at subsequent mentions.

Maximum noun phrases are the unit on which NP rewrite operates. They are defined in a dependency parse tree as the subtree that has as a root a noun such that there is no other noun on the path between it and the root of the tree. For example , there are two maximum NPs, with heads "police" and "Augusto_Pinochet" in the sentence "British police arrested former Chilean dictator Augusto Pinochet". The noun phrase "former chilean dictator" is not a maximum NP, since there is a noun (augusto_pinochet) on the path in the dependency tree between the noun "dictator" and the root of the tree. By definition a maximum NP includes all nominal and adjectival premodifiers of the head, as well as postmodifiers such as prepositional phrases, appositions, and relative clauses. This means that maximum NPs can be rather complex, covering a wide range of production rules in a context-free grammar. The dependency tree definition of maximum noun phrase makes it easy to see why these are a good unit for subsentential rewrite: the subtree that has the head of the NP as a root contains only modifiers of the head, and by rewriting the noun phrase, the amount of information expressed about the head entity can be varied.

In our implementation, a context free grammar probabilistic parser (Charniak, 2000) was used to parse the input. The maximum noun phrases were identified by finding sequences of <*np*>...</*np*> tags in the parse such that the number of opening and closing tags is equal. Each NP identified by such tag spans was considered as a candidate for rewrite.

Coreference classes A coreference class CR_m is the class of all maximum noun phrases in the input that refer to the same entity E_m. The general problem of coreference resolution is hard, and is even more complicated for the multi-document summarization case, in which cross-document resolution needs to be performed. Here we make a simplifying assumption, stating that all noun phrases that have the same noun as a head belong to the same coreference class. While we expected that this assumption would lead to some wrong decisions, we also suspected that in most common summarization scenarios, even if there are more than one entities expressed with the same noun, only one of them would be the *main* focus for the news story and will appear more often across input sentences. References to such main entities will be likely to be picked in a sentence for inclusion in the summary by chance more often than other competeing entities. We thus used the head noun equivalance to form the classes. A post-evaluation inspection of the summaries confirmed that our assumption was correct and there were only a small number of errors in the rewritten summaries that were due to coreference errors, which were greatly outnumbered by parsing errors for example. In a future evaluation, we will evaluate the rewrite module assuming perfect coreference and parsing, in order to see the impact of the core NP-rewrite approach itself.

3 NP rewrite evaluation

The NP rewrite summarization algorithm was applied to the 50 test sets for generic multi-document

summarization from the 2004 Document Understanding Conference. Two examples of its operation with $F_{RW} \equiv Avr$ are shown below.

Original.1 While the British government defended *the arrest*, it took no stand on extradition of Pinochet to Spain.

NP-Rewite.1 While the British government defended *the arrest in London of former Chilean dictator Augusto Pinochet*, it took no stand on extradition of Pinochet to Spain.

Original.2 *Duisenberg* has said growth in the euro area countries next year will be about 2.5 percent, lower than *the 3 percent* predicted earlier.

NP-Rewrite.2 *Wim Duisenberg, the head of the new European Central Bank,* has said growth in the euro area will be about 2.5 percent, lower than *just 1 percent in the euro-zone unemployment* predicted earlier.

We can see that in both cases, the NP rewrite pasted into the sentence important additional information. But in the second example we also see an error that was caused by the simplifying assumption for the creation of the coreference classes according to which the percentage of unemployment and growth have been put in the same class.

In order to estimate how much the summary is changed because of the use of the NP rewrite, we computed the unigram overlap between the original extractive summary and the NP-rewrite summary. As expected, $F_{FW} \equiv Sum$ leads to bigger changes and on average the rewritten summaries contained only 54% of the unigrams from the extractive summaries; for $F_{RW} \equiv Avr$, there was a smaller change between the extractive and the rewritten summary, with 79% of the unigrams being the same between the two summaries.

3.1 Linguistic quality evaluation

Noun phrase rewrite has the potential to improve the referential clarity of summaries, by inserting in the sentences more information about entities when such is available. It is of interest to see how the rewrite version of the summarizer would compare to the extractive version, as well as how its linguistic quality compares to that of other summarizers that participated in DUC. Four summarizers were evaluated: peer 117, which was a system that used generation techniques to produce the summary and was the only real non-extractive summarizer participant at DUC 2004 (Vanderwende et al., 2004); the extractive frequency summarizer, and the two versions of the rewrite algorithm (Sum and Avr). The evaluated rewritten summaries had potential errors coming from different sources, such as coreference resolution, parsing errors, sentence splitting errors, as well as errors coming directly from rewrite, in which an unsuitable NP is chosen to be included in the summary. Improvements in parsing for example could lead to better overall rewrite results, but we evaluated the output as is, in order to see what is the performance that can be expected in a realistic setting for fully automatic rewrite.

SYSTEM	$\mathbf{Q}_1$	$\mathbf{Q}_2$	$\mathbf{Q}_3$	$\mathbf{Q}_4$	$\mathbf{Q}_5$
SUM$_{Id}$	4.06	4.12	3.80	3.80	3.20
SUM$_{Avr}$	3.40	3.90	3.36	3.52	2.80
SUM$_{Sum}$	2.96	3.34	3.30	3.48	2.80
peer 117	2.06	3.08	2.42	3.12	2.10

Table 1: Linguistic quality evaluation. Peer 117 was the only non-extractive system entry in DUC 2004; SUM$_{Id}$ is the frequency summarizer with no NP rewrite; and the two versions of rewrite with sum and average as combination functions.

The evaluation was done by five native English speakers, using the five DUC linguistic quality questions on grammaticality (Q_1), repetition (Q_2), referential clarity (Q_3), focus (Q_4) and coherence (Q_5). Five evaluators were used so that possible idiosyncratic preference of a single evaluator could be avoided. Each evaluator evaluated all five summaries for each test set, presented in a random order. The results are shown in table 3.1. Each summary was evaluated for each of the properties on a scale from 1 to 5, with 5 being very good with respect to the quality and 1, very bad.

Comparing NP rewrite to extraction Here we would be interested in comparing the extractive frequency summarizer (SUM$_{Id}$), and the two version of systems that rewrite noun phrases: SUM$_{Avr}$ (which changes about 20% of the text) and SUM$_{Sum}$ (which changes about 50% of the text). The general trend that we see for all five dimensions of linguistic quality is that the more the text is automatically altered, the worse the linguistic quality of the summary

gets. In particular, the grammaticality of the summaries drops significantly for the rewrite systems. The increase of repetition is also significant between SUM_{Id} and SUM_{Sum}. Error analysis showed that sometimes increased repetition occurred in the process of rewrite for the following reason: the context weight update for words is done only after each noun phrase in the sentence has been rewritten. Occasionally, this led to a situation in which a noun phrase was augmented with information that was expressed later in the original sentence. The referential clarity of rewritten summaries also drops significantly, which is a rather disappointing result, since one of the motivations for doing noun phrase rewrite was the desire to improve referential clarity by adding information where such is necessary. One of the problems here is that it is almost impossible for human evaluators to ignore grammatical errors when judging referential clarity. Grammatical errors decrease the overall readability and a summary that is given a lower grammaticality score tends to also receive lower referential clarity score. This fact of quality perception is a real challenge for summarization systems that move towards abstraction and alter the original wording of sentences since certainly automatic approaches are likely to introduce ingrammaticalities.

Comparing SUM_{Sum} and peer 117 We now turn to the comparison of between SUM_{Sum} and the generation based system 117. This system is unique among the DUC 2004 systems, and the only one that year that experimented with generation techniques for summarization. System 117 is verb-driven: it analizes the input in terms of predicate-argument triples and identifies the most important triples. These are then verbalized by a generation system originally developed as a realization component in a machine translation engine. As a result, peer 117 possibly made even more changes to the original text then the NP-rewrite system. The results of the comparison are consistent with the observation that the more changes are made to the original sentences, the more the readability of summaries decreases. SUM_{Sum} is significantly better than peer 117 on all five readability aspects, with notable difference in the grammaticality and referential quality, for which SUM_{Sum} outperforms peer 117 by a full point. This indicates that NPs are a good candidate granularity for sentence changes and it can lead to substantial altering of the text while preserving significantly better overall readability.

3.2 Content selection evaluation

We now examine the question of how the content in the summaries changed due to the NP-rewrite, since improving content selection was the other motivation for exploring rewrite. In particular, we are interested in the change in content selection between SUM_{Sum} and SUM_{Id} (the extractive version of the summarizer). We use SUM_{Sum} for the comparison because it led to bigger changes in the summary text compared to the purely extractive version. We used the pyramid evaluation method: four human summaries for each input were manually analyzed to identify shared *content units*. The weight of each content unit is equal to the number of model summaries that express it. The pyramid score of an automatic summary is equal to the weight of the content units expressed in the summary divided by the weight of an ideally informative summary of the same length (the content unit identification is again done manually by an annotator).

Of the 50 test sets, there were 22 sets in which the NP-rewritten version had lower pyramid scores than the extractive version of the summary, 23 sets in which the rewritten summaries had better scores, and 5 sets in which the rewritten and extractive summaries had exactly the same scores. So we see that in half of the cases the NP-rewrite actually improved the content of the summary. The summarizer version that uses NP-rewrite has overall better content selection performance than the purely extractive system. The original pyramid score increased from 0.4039 to 0.4169 for the version with rewrite. This improvement is not significant, but shows a trend in the expected direction of improvement.

The lack of significance in the improvement is due to large variation in performance: when np rewrite worked as expected, content selection improved. But on occasions when errors occurred, both readability and content selection were noticeably compromised. Here is an example of summaries for the same input in which the NP-rewritten version had better content. After each summary, we list the content units from the pyramid content analysis that were expressed in the summary. The weight of each

content unit is given in brackets before the label of the unit and content units that differ between the extractive and rewritten version are displayed in italic. The rewritten version conveys high weight content units that do not appear in the extractive version, with weights 4 (maximum weight here) and 3 respectively.

Extractive summary Italy's Communist Refounding Party rejected Prime Minister Prodi's proposed 1999 budget. By one vote, Premier Romano Prodi's center-left coalition lost a confidence vote in the Chamber of Deputies Friday, and he went to the presidential palace to rsign. Three days after the collapse of Premier Romano Prodi's center-left government, Italy's president began calling in political leaders Monday to try to reach a consensus on a new government. Prodi has said he would call a confidence vote if he lost the Communists' support." I have always acted with coherence," Prodi said before a morning meeting with President Oscar Luigi.

(4) Prodi lost a confidence vote

(4) The Refounding Party is Italy's Communist Party

(4) The Refounding Party rejected the government's budget

(3) The dispute is over the 1999 budget

(2) Prodi's coalition was center-left coalition

(2) The confidence vote was lost by only 1 vote

(1) Prodi is the Italian Prime Minister

(1) Prodi wants a confidence vote from Parliament

NP-rewrite version Communist Refounding, a fringe group of hard-line leftists who broke with the minstream Communists after they overhauled the party following the collapse of Communism in Eastern Europe rejected Prime Minister Prodi's proposed 1999 budget. By only one vote, the center-left prime minister of Italy, Romano Prodi, lost The vote in the lower chamber of Parliament 313 against the confidence motion brought by the government to 312 in favor in Parliament Friday and was toppled from power. President Oscar Luigi Scalfaro, who asked him to stay on as caretaker premier while the head of state decides whether to call elections.

(4) Prodi lost a confidence vote

(4) Prodi will stay as caretaker until a new government is formed

(4) The Refounding Party is Italy's Communist Party

(4) The Refounding Party rejected the government's budget

(3) Scalfaro must decide whether to hold new elections

(3) The dispute is over the 1999 budget

(2) Prodi's coalition was center-left coalition

(2) The confidence vote was lost by only 1 vote

(1) Prodi is the Italian Prime Minister

Below is another example, showing the worse deterioration of the rewritten summary compared to the extractive one, both in terms of grammaticality and content. Here, the problem with repetition during rewrite arises: the same person is mentioned twice in the sentence and at both places the same overly long description is selected during rewrie, rendering the sentence practically unreadable.

Extractive summary Police said Henderson and McKinney lured Shepard from the bar by saying they too were gay and one of their girlfriends said Shepard had embarrassed one of the men by making a pass at him. 1,000 people mourned Matthew Shepherd, the gay University of Wyoming student who was severely beaten and left to die tied to a fence. With passersby spontaneously joining the protest group, two women held another sign that read," No Hate Crimes in Wyoming." Two candlelight vigils were held Sunday night. Russell Anderson, 21, and Aaron McKinney, 21, were charged with attempted murder.

(4) The victim was a student at the University of Wyoming

(4) The victim was brutally beaten

(4) The victim was openly gay

(3) The crime was widely denounced

(3) The nearly lifeless body was tied to a fence

(3) The victim died

(3) The victim was left to die

(2) The men were arrested on charges of kidnapping and attempted first degree murder

(2) There were candlelight vigils in support for the victim

(1) Russell Henderson and Aaron McKinney are the names of the people responsible for the death

NP-rewrite version Police said Henderson and McKinney lured the The slight, soft-spoken 21-year-old Shepard, a freshman at the University of Wyoming, who became an overnight symbol of anti-gay violence after he was found dangling from the fence by a passerby from a bar by saying they too were gay and one of their girlfriends said the The slight, soft-spoken 21-year-old Shepard, a freshman at the University of Wyoming, who became an overnight symbol of anti-gay violence after he was found dangling from the fence by a passerby had embarrassed one of the new ads in that supposedly hate-free crusade.

(4) The victim was a student at the University of Wyoming

(3)The nearly lifeless body was tied to a fence *(1) A passerby found the victim*

(1) Russell Henderson and Aaron McKinney are the names of the people responsible for the death

(1) The victim was 22-year old

Even from this unsuccessful attempt for rewrite we can see how changes of the original text can be desirable, since some of the newly introduced information is in fact suitable for the summary.

4 Conclusions

We have demonstrated that an entity-driven approach to rewrite in multi-document summarization can lead to considerably different summary, in terms of content, compared to the extractive version of the same system. Indeed, the difference leads to some improvement measurable in terms of pyramid method evaluation. The approach also significantly outperforms in linguistic quality a non-extractive event-centric system.

Results also show that in terms of linguistic quality, extractive systems will be curently superior to systems that alter the original wording from the input. Sadly, extractive and abstractive systems are evaluated together and compared against each other, putting pressure on system developers and preventing them from fully exploring the strengths of generation techniques. It seems that if researchers in the field are to explore non-extractive methods, they would need to compare their systems separately from extractive systems, at least in the beginning exploration stages. The development of non-extractive approaches in absolutely necessary if automatic summarization were to achieve levels of performance close to human, given the highly abstractive form of summaries written by people.

Results also indicate that both extractive and non-extractive systems perform rather poorly in terms of the focus and coherence of the summaries that they produce, identifying macro content planning as an important area for summarization.

References

Regina Barzilay and Kathleen McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3).

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *NAACL-2000*.

John Conroy, Judith Schlesinger, and Dianne O'Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of ACL, companion volume*.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, Sydney, Australia.

Gunes Erkan and Dragomir Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

Hongyan Jing and Kathleen McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization — step one: Sentence compression. In *Proceeding of The American Association for Artificial Intelligence Conference (AAAI-2000)*, pages 703–710.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics*, pages 495–501.

Chin-Yew Lin and Eduard Hovy. 2002. Automated multi-document summarization in neats. In *Proceedings of the Human Language Technology Conference (HLT2002)*.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

K. McKeown, R. Passonneau, D. Elson, A. Nenkova, and J. Hirschberg. 2005. Do summaries help? a task-based evaluation of multi-document summarization. In *SIGIR*.

R. Mihalcea and P. Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004*, pages 404–411.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of SIGIR*.

Barry Schiffman, Ani Nenkova, and Kathleen McKeown. 2002. Experiments in multidocument summarization. In *Proceedings of the Human Language Technology Conference*.

Lucy Vanderwende, Michele Banko, and Arul Menezes. 2004. Event-centric summary generation. In *Proceedings of the Document Understanding Conference (DUC'04)*.

A New Approach to Automatic Document Summarization

Xiaofeng Wu
National Laboratory of Pattern Recognition,
Institute of Automation,
Chinese Academy of Sciences
Beijing, China
xfwu@nlpr.ia.ac.cn

Chengqing Zong
National Laboratory of Pattern Recognition,
Institute of Automation,
Chinese Academy of Sciences
Beijing, China
cqzong@nlpr.ia.ac.cn

Abstract

In this paper we propose a new approach based on *Sequence Segmentation Models* (SSM) to the extractive document summarization, in which summarizing is regarded as a segment labeling problem. Comparing with the previous work, the difference of our approach is that the employed features are obtained not only from the sentence level, but also from the segment level. In our approach, the semi-Markov CRF model is employed for segment labeling. The preliminary experiments have shown that the approach does outperform all other traditional supervised and unsupervised approaches to document summarization.

1 Introduction

Document summarization has been a rapidly evolving subfield of Information Retrieval (IR) since (Luhn, 1958). A summary can be loosely defined as a text that is produced from one or more texts and conveys important information of the original text(s). Usually it is no longer than half of the original text(s) or, significantly less (Radev et al., 2002). Recently, many evaluation competitions (like the Document Understanding Conference DUC "http://duc.nist.gov", in the style of NIST's TREC), provided some sets of training corpus. It is obvious that, in the age of information explosion, document summarization will be greatly helpful to the internet users; besides, the techniques it uses can also find their applications in speech techniques and multimedia document retrieval, etc.

The approach to summarizing can be categorized in many ways. Some of them are: 1) *indicative*, *informative* and *evaluative*, according to functionality; 2) *single-document* and *multi-document*, according to the amount of input documents; 3) *generic* and *query-oriented*, according to applications. Yet the taxonomy currently widely employed is to categorize summarization into *abstractive* and *extractive*.

According to (Radev et al., 2002), all methods that are not explicitly extractive are categorized as abstractive. These approaches include ontological information, information fusion, and compression. Abstract-based summarization never goes beyond conceptual stage, though ever since the dawn of summarization it has been argued as an alternative for its extract-based counterpart. On the other hand, extractive summarization is still attracting a lot of researchers (Yeh et al., 2005) (Daum´e III and Marcu, 2006) and many practical systems, say, MEAD "http://www.summarization.com/mead/", have been produced. Using supervised or unsupervised machine learning algorithms to extract sentences is currently the mainstream of the extractive summarization. However, all pervious methods focus on obtaining features from the sentence granularity.

In this paper we focus on generating summarization by using a supervised extractive approach in which the features are obtained from a larger granularity, namely segment. The remainder of the paper is organized as follows: Section 2 introduces the related work concerning the extract-based summarization. Section 3 describes our motivations. Our experiments and results are given in Section 4, and Section 5 draws the conclusion and mentions the future work.

2 Related Work

Early researchers approached the summarization problem by scoring each sentence with a combination of the features like word frequency and distribution, some proper names (Luhn, 1958), sentence positions in a paragraph (Baxendale, 1958), and sentence similarity (Gong, 2001) etc. The results were comparatively good. Most supervised extractive methods nowadays focus on finding powerful machine learning algorithms that can properly combine these features.

Bayesian classifier was first applied to summarization by (Pedersen and Chen, 1995), the authors claimed that the corpus-trained feature weights were in agreement with (Edmundson, 1969), which employed a subjective combination of weighted features. Another usage of the naïve Bayesian model in summarization can be found in (Aone et al., 1997). Bayesian model treats each sentence individually, and misses the intrinsic connection between the sentences. (Yeh et al., 2005) employed genetic algorithm to calculate the belief or score of each sentence belonging to the summary, but it also bears this shortcoming.

To overcome this independence defect, (Conroy and O'leary, 2001) pioneered in deeming this problem as a *sequence labeling problem*. The authors used HMM, which has fewer independent assumptions. However, HMM can not handle the rich linguistic features among the sentences either. Recently, as CRF (Lafferty and McCallum, 2001) has been proved to be successful in part-of-speech tagging and other sequence labeling problems, (Shen et al., 2007) attempted to employ this model in document summarization. CRF can leverage all those features despite their dependencies, and absorb other summary system's outcome. By introducing proper features and making a comparison with SVM, HMM, etc., (Shen et al., 2007) claimed that CRF could achieve the best performance.

All these approaches above share the same viewpoint that features should be obtained at sentence level. Nevertheless, it can be easily seen that the non-summary or summary sentences tend to appear in a consecutive manner, namely, in segments. These rich features of segments can surely not be managed by those traditional methods.

Recently, Sequence Segmentation Model (SSM) has attracted more and more attention in some traditional sequence learning tasks. SSM builds a direct path to encapsulate the rich segmental features (e.g., entity length and the similarity with other entities, etc., in entity recognition). Semi-CRF (Sarawagi and Cohen, 2004) is one of the SSMs, and generally outperforms CRF.

3 Motivations

According to the analysis in Section 2, our basic idea is clear that we regard the supervised summarizing as a *problem of sequence segmentation*. However, in our approach, the features are not only obtained on the sentence level but also on the segment level.

Here a segment means one or more sentences sharing the same label (namely, non-summary or summary), and a text is regarded as a sequence of segments. Semi-CRF is a qualified model to accomplish the task of segment labeling, besides it shares all the virtues of CRF. Using semi-CRF, we can easily leverage the features both in traditional sentence level and in the segment level. Some features, like Log Likelihood or Similarity, if obtained from each sentence, are inclined to give unexpected results due to the small granularity. Furthermore, semi-CRF is a generalized version of CRF. The features designed for CRF can be used in semi-CRF directly, and it has been proved that semi-CRF outperforms CRF in some Natural Language Processing (NLP) problems (Sarawagi and Cohen, 2004).

In the subsections below, we first introduce semi-CRF then describe the features we used in our approach.

3.1 Semi-CRF

CRF was first introduced in (Lafferty and McCallum, 2001). It is a conditional model $P(Y|X)$, and here both X and Y may have complex structure. The most prominent merits of CRF are that it offers relaxation of the strong independence assumptions made in HMM or Maximum Entropy Markov Models (MEMM) (McCallum, 2000) and it is no victim of the label bias problem. Semi-CRF is a generalization version of sequential CRF. It extends CRF by allowing each state to persist for a non-unit length of time. After this time has elapsed, the system might transmit to a new state, which only depends on its previous one. When the system is in the "segment of time", it is allowed to behave non-Markovianly.

3.1.1 CRF vs. Semi-CRF

Given an observed sentence sequence $X=(x_1,x_2,\ldots,x_M)$. The corresponding output labels are $Y=(y_1,y_2,\ldots,y_M)$, where y_i gets its value from a fixed set Ψ. For document summarization, $\Psi=\{0,1\}$. Here 1 for summary and 0 for non-summary. The goal of CRF is to find a sequence of Y, that maximize the probability:

$$P(Y \mid X,W) = \frac{1}{Z(X)} \exp(W \cdot F(X,Y)) \quad (1)$$

Here, $F(X,Y) = \sum_{i=1}^{M} \mathrm{f}(i,X,Y)$ is a vertical vector of size T. The vertical vector $\mathrm{f} = (f_1, f_2, ..., f_T)'$ means there are T feature functions, and each of them can be written as $f_t(i,X,Y) \in R, t \in (1,\cdots,T), i \in (1,\cdots,M)$. For example, in our experiment the 10th feature function is expressed as: [if the length of current sentence is bigger than the predefined threshold value]&[if the current sentence is a summary]. When this feature function is acting upon the third sentence in *text*_1 with *label_sequence*_1, the following feature equation f_{10}(3,*text*_1, *label_sequence*_1) means: in *text*_1 with *label_sequence*_1, [if the length of the third sentence is bigger than the predefined threshold value]&[if the third sentence is a summary]. W is a horizontal vector of size T that represents the weights of these features respectively. Equation (2) gives the definition of $Z(X)$, which is a normalization constant that makes the probabilities of all state sequences sum to 1.

$$Z(X) = \sum_{Y'} \exp(W \cdot F(X,Y')) \quad (2)$$

If we change the sequence vector X to $S=<s_1,s_2,\ldots,s_N>$, which means *one way to* split X into N segments, we have the semi-CRF. Each element in S is a triple: $S_j=<t_j,u_j,y_j>$, which denotes the j^{th} segment in this way of segmentation. In the triple, t_j denotes the start point of the j^{th} segment, u_j denotes its end position, and y_j is the output label of the segment (recall the example at the beginning of this subsection that there is only one output for a segment). Under this definition, segments should have no overlapping, and satisfy the following conditions:

$$\sum_{j=1}^{N} |s_j| = |X| \quad (3)$$

$$t_1 = 1, u_N = |X|, 1 \le t_j \le u_j \le |X|, t_{j+1} = u_j + 1 \quad (4)$$

Here, $|\bullet|$ denotes the length of $\bullet$.

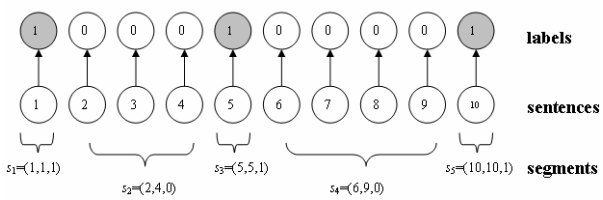


Figure 1 A 10-sentences text with label sequence

For example, one way to segment a text of 10 sentences in Figure 1 is $S=<(1,1,1),(2,4,0),(5,5,1),(6,9,0),(10,10,1)>$. The circles in the second row represent sentences, and actually are only some *properties* of the corresponding sentences.

Consequently, the feature function f in CRF converts to the segmental feature function $\mathrm{g}=(g_1,g_2,\ldots,g_{T'})$. Like f, $g_t(i,x,s) \in R$ also maps a triple (i,x,s) to a real number. Similarly, we may define $G(X,S) = \sum_{i=1}^{N} \mathrm{g}(i,X,S)$. Now we give the final equation used to estimate the probability of S. Given a sequence X and feature weight W, we have

$$P(S \mid X,W) = \frac{1}{Z(X)} \exp(W \cdot G(X,S)) \quad (5)$$

Here,

$$Z(X) = \sum_{S' \in \Delta} \exp(W \cdot G(X,S')) \quad (6)$$

Where, $\Delta = \{all - segmentations - allowed\}$.

3.1.2 Inference

The inference or the testing problem of semi-CRF is to find the best S that maximizes Equation (5). We use the following Viterbi-like algorithm to calculate the optimum path.

Suppose the longest segment in corpus is K, let $S_{1:i,y}$ represent all possible segmentations starting from 1 to i, and the output of the last segment is y. $V(i,y)$ denotes the biggest value of $P(S'|X,W)$. Note that it's also the largest value of $W \cdot G(X,S')$, $S' \in S_{1:i,y}$.

Compared with the traditional Viterbi algorithm used in CRF, the inference for semi-CRF is more time-consuming. But by studying Algorithm 1, we can easily find out that the cost is only linear in K.

Algorithm 1:

Step1. Initialization:

Let $V(i, y) = 0$, *for* $i = 0$

Step2. Induction:

for $i > 0$

$$V(i, y) = \max_{y', k=1,\ldots,K} V(i-k, y') + W \cdot \mathrm{g}(y, y', x, i-d+1, i) \quad (7)$$

Step3. Termination and path readout:

$$bestSegment = \max_y V(|X|, y)$$

3.1.3 Parameter Estimation

Define the following function

$$L_W = \sum_l \log P(S_l \mid X_l, W) = \sum_l (W \cdot G(X_l, S_l) - \log Z(X_l)) \quad (8)$$

In this approach, the problem of parameter estimation is to find the best weight W that maximizes L_W. According to (Bishop, 2006), the Equation (8) is convex. So it can be optimized by gradient ascent. Various methods can be used to do this work (Pietra et al. 1997). In our system, we use L-BFGS, a quasi-Newton algorithm (Liu and Nocedal. 1989), because it has the fast converging speed and efficient memory usage. APIs we used for estimation and inference can be found in website "http:\\crf.sourcefourge.net".

3.2 Features

(Shen et al. 2007) has made a thorough investigation of the performances of CRF, HMM, and SVM. So, in order to simplify our work and make it comparable to the previous work, we shape our designation of features mainly under their framework.

The mid column in Table 1 lists all of the features we used in our semi-CRF approach. For the convenience of comparison, we also list the name of the features used in (Shen et al. 2007) in the right column, and name them *Regular Features*. The features in bold-face in the mid column are the corresponding features tuned to fit for the usage of semi-CRF. We name them *Extended Features*. There are some features that are not in bold-face in the mid column. These features are the same as the *Regular Features* in the right column. We also used them in our approach. The mark star denotes that there is no counterpart. We number these features in the left column.

No.	semi-CRF	CRF
1	**Ex_Position**	Position
2	**Ex_Length**	Length
3	**Ex_Log_Likelihood**	Log Likelihood
4	**Ex_Similarity_to_ Neighboring_ Segments**	Similarity to Neighboring Sentences
5	**Ex_Segment_ Length**	*
6	*Thematic*	Thematic
7	*Indicator*	Indicator
8	*Upper Case*	Upper Case

Table 1. *Features List*

The details of the features we used in semi-CRF are explained as follow.

Extended Features:

Ex_Position: is an extended version of the Position feature. It gives the description of the position of a segment in the current segmentation. If the sentences in the current segment contain the beginning sentence of a paragraph, the value of this feature will be 1, 2 if it contains the end of a paragraph; and 3 otherwise;

Ex_Length: the number of words in the current segment after removing some stop-words.

Ex_Log_Likelihood: the log likelihood of the current segment being generated by the document. We use Equation (9) below to calculate this feature. $N(w_j,s_i)$ denotes the number of occurrences of the word w_j in the segment s_i, and we use $N(w_j, D)/\sum_{w_k} N(w_k, D)$ to estimate the probability of a word being generated by a document.

$$\log P(s_i \mid D) = \sum_{w_j} N(w_j, s_i) \log p(w_j \mid D) \quad (9)$$

Ex_Similarity_to_Neighboring_Segments: we define the cosine similarity based on the TF*IDF (Frakes &Baeza-Yates, 1992) between a segment and its neighbors. But unlike (Shen et al. 2007), in our work only the adjacent neighbors of the segment in our work are considered.

EX_Segment_Length: this feature describes the number of sentences contained in a segment.

All these features above are actually an extended version used in the regular CRF (or in other supervised model). It is easy to see that, if the segment length is equal to 1, then the features will degrade to their normal forms.

There are some features that are also used in semi-CRF but we don't extend them like those features above. Because the extended version of these features leads to no improvement of our result. These features are:

Regular features we used:

Thematic: with removing of stop words, we define the words with the highest frequency in the document to be the thematic words. And this feature gives the count of these words in each sentence.

Indicator: indicative words such as "conclusion" and "briefly speaking" are very likely to be included in summary sentences, so we define this feature to signal if there are such words in a sentence.

Upper Case: some words with upper case are of high probability to be a name, and sentences with such words together with other words which the author might want to emphasize are likely to be appeared in a summary sentence. So we use this feature to indicate whether there are such words in a sentence.

It should be noted that theoretically the number of extended features obtained from the corpus goes linearly with K in Equation (7).

4 Experiments

4.1 Corpus & Evaluation Criteria

To evaluate our approach, we applied the widely used test corpus of (DUC2001), which is sponsored by ARDA and run by NIST "http://www.nist.gov". The basic aim of DUC 2001 is to further progress of summarization and enable researchers to participate into large-scale experiments. The corpus DUC2001 we used contains 147 news texts, each of which has been labeled manually whether a sentence belongs to a summary or not. Because in (Shen et al. 2007) all the experiments were conducted upon DUC2001, we may make a comparison between the *sequence labeling models* and the *sequence segmentation modes* we used. The only preprocessing we did is to remove some stop words according to a stop word list.

We use $F1$ score as the evaluation criteria which is defined as:

$$F1 = \frac{2*\text{Precesion}*\text{Recall}}{\text{Precesion}+\text{Recall}} \qquad (10)$$

We used 10-fold cross validation in order to reduce the uncertainty of the model we trained. The final $F1$ score reported is the average of all these 10 experiments.

All those steps above are strictly identical to the work in (Shen et al. 2007), and its result is taken as our baseline.

4.2 Results & Analysis

As we mentioned in Sub-Section 3.2, those extended version of features only work when segment length is bigger than one. So, each of these extended version of features or their combination can be used together with all the other regular features listed in the right column in Table 1. In order to give a complete test of the capacity of all these extended features and their combinations, we do the experiments according to the power set of {1, 2, 3, 4, 5} (the numbers are the IDs of these extended features as listed in Table 1), that is we need to do the test 2^5-1 times with different combinations of the extended features. The results are given in Table 2. The rows with italic fonts (1, 3, 5, 7, 9, 11, 13), in Table 2 denote the extended features used. For example, '*1+2*' means that the features Ex_Positon and the Ex_Length are **together used with** all other regular features are used.

	1	*2*	*3*	*4*	*5*
F1	0.395	0.391	0.398	0.394	0.392
	1+2	*1+3*	*1+4*	*1+5*	*2+3*
F1	0.395	0.396	0.396	0.395	0.382
	2+4	*2+5*	*3+4*	*3+5*	*4+5*
F1	0.389	0.384	0.398	0.399	0.380
	1+2+3	*1+2+4*	*1+2+5*	*1+3+4*	*1+3+5*
F1	0.398	0.397	0.393	0.403	0.402
	1+4+5	*2+3+4*	*2+3+5*	*2+4+5*	*3+4+5*
F1	0.402	0.403	0.401	0.403	0.404
	1+2 +3+4	*1+2 +3+5*	*1+2 +4+5*	*1+3 +4+5*	*2+3 +4+5*
F1	**0.407**	0.404	0.406	0.402	0.404
	All	*CRF*			
F1	0.406	0.389			

Table 2. *Experiment results.*

Other rows (2, 4, 6, 8, 10, 12, 14) give *F*1 scores corresponding to the features used.

In Table 3 we compare our approach with some of the most popular unsupervised methods, including LSA (Frakes & Baeza-Yates, 1992) and HITS (Mihalcea 2005). The experiments were conducted by (Shen et al. 2007).

	LSA	*HITS*	*Seim-CRF*
F1	0.324	0.368	0.407

Table 3 *Comparison with unsupervised methods*

From the results in Table 2 we can see that individually applying these extended features can improve the performance somewhat. The best one of these extended features is feature 3, as listed in the 2nd row, the 5th column. The highest improvement, 1.8%, is obtained by combining the features 1, 2, 3 and 4. Although a few of the combinations hurt the performance, most of them are helpful. This verifies our hypothesis that the extended features under SSM have greater power than the regular features. The results in Table 3 demonstrate that our approach significantly outperforms the traditional unsupervised methods. 8.3% and 4.9% improvements are respectively gained comparing to LSA and HITS models

Currently, the main problem of our method is that the searching space goes large by using the extended features and semi-CRF, so the training procedure is time-consuming. However, it is not so unbearable, as it has been proved in (Sarawagi and Cohen, 2004).

5 Conclusion and Future Work

In this paper, we exploit the capacity of semi-CRF , we also make a test of most of the common features and their extended version designed for document summarization. We have compared our approach with that of the regular CRF and some of the traditional unsupervised methods. The comparison proves that, because summary sentences and non-summary sentences are very likely to show in a consecutive manner, it is more nature to obtain features from a lager granularity than sentence.

In our future work, we will test this approach on some other well known corpus, try the complex features used in (Shen et al. 2007), and reduce the time for training.

Acknowledgements

The research work described in this paper has been funded by the Natural Science Foundation of China under Grant No. 60375018 and 60121302.

References

C.Aone, N. Charocopos, J. Gorlinsky. 1997. An Intelligent Multilingual Information Browsing and Retrieval System Using Information Extraction. In *ANLP*, 332-339.

P.B. Baxendale. 1958. Man-made Index for Technical Literature -An Experiment. *IBM Journal of Research and Development*, 2(4):354-361.

C.M. Bishop. 2006. Linear Models for Classification, *Pattern Recognition and Machine Learning*, *chapter 4*, Springer.

J. M. Conroy and D. P. O'leary. 2001. Text Summarization via Hidden Markov Models. In *SIGIR*, 406-407.

Hal Daum´e III, and D. Marcu. 2006. Bayesian Query- Focused Summarization, In *ACL*

H. P. Edmundson. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, 16(2):264-285.

W. B. Frakes, R. Baeza-Yates, 1992, *Information Retrieval Data Structures & Algorithms.* Prentice Hall PTR, New Jersey

Y. H. Gong and X. Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *SIGIR,* 19-25

J. Kupiec, J. Pedersen, and F. Chen. 1995. A Trainable Document Summarizer. *Research and Development in Information Retrieval,* 68-73

J. D. Lafferty, A. McCallum and F. C. N. Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. *ICML*, 282-289.

D. C. Liu and J. Nocedal. 1989. On the limited memory BFGS method for large-scale optimization. *Mathematic Programming*, 45:503-528.

H. P. Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2): 159 -165.

A. McCallum, D. Freitag, and F. Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *ICML*, 591-598

Mihalcea R. Mihalcea. 2005. Language independent extractive summarization. In *AAAI*, 1688-1689

S. D. Pietra, V. D. Pietra, and J. D. Lafferty. 1997. Inducing features of random fields. IEEE Tran. *on Pattern Analysis and Machine Intelligence*, 19(:)380–393.

D. R. Radev, E. Hovy and K. McKeown. 2002. Introduction to the Special Issue on Summarization. *Computational Linguistics*, 28(4): 399-408.

S. Sarawagi and W.W. Cohen. 2004. Semi-markov conditional random fields for information extraction.In *NIPS*

D. Shen, J. T. Sun, H. Li, Q. Yang, Z. Chen. 2007. Document Summarization using Conditional Random Fields' In *IJCAI*, 1805-1813

J. Y. Yeh, H. R. Ke, W. P. Yang and I. H. Meng. 2005. Text summarization using trainable summarizer and latent semantic analysis. *IPM*, 41(1): 75–95

Generic Text Summarization Using Probabilistic Latent Semantic Indexing

Harendra Bhandari
Graduate School of Information Science
Nara Institute of Science and Technology
Nara 630-0192, Japan
harendra-b@is.naist.jp

Masashi Shimbo
Graduate School of Information Science
Nara Institute of Science and Technology
Nara 630-0192, Japan
shimbo@is.naist.jp

Takahiko Ito
Graduate School of Information Science
Nara Institute of Science and Technology
Nara 630-0192, Japan
takahahi-i@is.naist.jp

Yuji Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
Nara 630-0192, Japan
matsu@is.naist.jp

Abstract

This paper presents a strategy to generate generic summary of documents using Probabilistic Latent Semantic Indexing. Generally a document contains several topics rather than a single one. Summaries created by human beings tend to cover several topics to give the readers an overall idea about the original document. Hence we can expect that a summary containing sentences from better part of the topic spectrum should make a better summary. PLSI has proven to be an effective method in topic detection. In this paper we present a method for creating extractive summary of the document by using PLSI to analyze the features of document such as term frequency and graph structure. We also show our results, which was evaluated using ROUGE, and compare the results with other techniques, proposed in the past.

1 Introduction

The advent of the Internet has made a wealth of textual data available to everyone. Finding a specific piece of information in this mass of data can be compared with "finding a small needle in a large heap of straw." Search engines do a remarkable job in providing a subset of the original data set which is generally a lot smaller than the original pile of data. However the subset provided by the search engines is still substantial in size. Users need to manually scan through all the information contained in the list of results provided by the search engines until the desired information is found. This makes automatic summarization the task of great importance as the users can then just read the summaries and obtain an overview of the document, hence saving a lot of time during the process.

Several methods have been proposed in the field of automatic text summarization. In general two approaches have been taken, extract-based summarization and abstract-based summarization. While extract-based summarization focuses in finding relevant sentences from the original document and using the exact sentences as a summary, abstract-based summaries may contain the words or phrases not present in the original document (Mani, 1999). The summarization task can also be classified as query-oriented or generic. The query-oriented summary presents text that contains information relevant to the given query, and the generic summarization method presents the summary that gives overall sense of the document (Goldstein et al, 1998). In this paper, we will focus on extract-based generic single-document summarization.

In the recent years graph based techinques have become very popular in automatic text summariza-

tion (Erkan and Radev, 2004), (Mihalcea, 2005). These techniques view each sentence as a node of a graph and the similarities between each sentences as the links between those sentences. Generally the links are retained only if the similarity values between the sentences exceed a pre-determined threshold value; the links are discarded otherwise. The sentences are then ranked using some graph ranking algorithms such as HITS (Kleinberg, 1998) or PageRank (Brin and Page, 1998) etc. However the graph ranking algorithms tend to give the highest ranking to the sentences related to one central topic in the document. So if a document contains several topics, these algorithms will only choose one central topic and rank the sentences related to those topic higher than any other topics, ignoring the importance of other topics present. This will create summaries that may not cover the overall topics of the document and hence cannot be considered generic enough. We will focus on that problem and present a way to create better generic summary of the document using PLSI (Hofmann 1999) which covers several topics in the document and is closer to the summaries created by human beings. The benchmarking done using DUC[2] 2002 data set showed that our technique improves over other proposed methods in terms of ROUGE[1] evaluation score.

2 Related Work

2.1 Maximal Marginal Relevance(MMR)

MMR is a summarization procedure based on vector-space model and is suited to generic summarization (Goldstein et al, 1999). In MMR the sentence are chosen according to the weighed combination of their general relevance in the document and their redundancy with the sentences already chosen. Both the relevance and redundancy are measured using cosine similarity. Relevance is the cosine similarity of a sentence with rest of the sentence in the document whereas redundancy is measured using cosine similarity between the sentence and the sentences already chosen for the summary.

2.2 Graph Based Summarization

The graph-based summarization procedure are be coming increasingly popular in recent years. Lex PageRank (Erkan and Radev, 2004) is one of such methods. LexPageRank constructs a graph where each sentence is a node and links are the similarities between the sentences. Similarity is measured using cosine similarity of the word vectors, and if the similarity value is more than certain threshold value the link is kept otherwise the links are removed. PageRank is an algorithm which has been successfully applied by Google search engine to rank the search results. Similarly PageRank is applied in LexPageRank to rank the nodes (or, sentences) of the resultant graph. A similar summarization method has been proposed by Mihalcea (2005).

Algorithms like HITS and PageRank calculate the principal eigenvector (hence find the principal community) of the matrix representing the graph. But as illustrated in Figure 1, another eigenvector which is slightly smaller than the principal eigenvector may exist. In documents, each community represented by the eigenvectors can be considered as a topic present in the document. As these algorithms tend to ignore the influence of eigenvectors other than largest one, the sentences related to topics other than a central one can be ignored, and creating the possibility for the inclusion of redundant sentences as well. This kind of summary cannot be considered as a generic one.

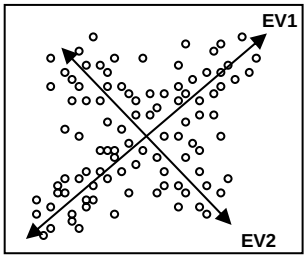


Figure 1. In algorithms like HITS and PageRank only the principal eigenvectors are considered. In the figure the vector EV1 is slightly larger than vector EV2, but the score commanded by members of EV2 communities are ignored.

As we mentioned in section 1, we take into consideration the sentences from all the topics generated by PLSI in the summary, hence getting a more generic summary.

2.3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Deerwester et al.,

[1] ROUGE:http://openrouge.com/default.aspx
[2] http://duc.nist.gov

1990) takes the high dimensional vector space representation of the document based on term frequency and projects it to lesser dimension space. It is thought that the similarities between the documents can be more reliably estimated in the reduced latent space representation than original representation. LSA has been applied in areas of text retrieval (Deerwester et al., 1990) and automatic text summarization (Gong and Liu, 2001). LSA is based on Singular Value Decomposition (SVD) of $m \times n$ term-document matrix A. Each entry in A, A_{ij}, represents the frequency of term i in document j. Using SVD, the matrix A is decomposed into U,S,V as,

$A=USV^T$

U=Matrix of n left singular vectors

S=diag(σ_i)=Diagonal matrix of singular values where with $\sigma_i \geq \sigma_{i+1}$ for all i.

V^T=Matrix of right singular vectors. Each row represents a topic and the values in each row represent the score of documents, represented by each columns, for the topic represented by the row.

Gong and Liu (2001) have proposed a scheme for automatic text summarization using LSA. Their algorithm can be stated below.

a. Choose the highest ranked sentence from k^{th} right singular vector in matrix V^T and use the sentence in summary.
b. If k reaches the predefined number, terminate the process; otherwise, go to step a again.

LSA categorizes sentences on the basis of the topics they belong to. Gong and Liu's method picks sentences from various topics hence producing the summaries that are generic in nature.

In section 3 we explain how PLSI is more advanced form of LSA. In section 5, we compare our summarization results with that of LSA.

3 Probabilistic Latent Semantic Indexing

Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999) is a new approach to automated document indexing, and is based on a statistical latent class model for factor analysis of count data. PLSI is considered to be a probabilistic analogue of Latent Semantic Indexing (LSI), which is a document indexing technique based on LSA. Despite the success of LSI, it is not devoid of deficits. The main argument against LSI is pointed to its unsatisfactory statistical foundations. In contrast, PLSI has solid statistical foundations, as it is based on the maximum likelihood principle and defines a proper generative model of data. Hofmann (1999) has shown that PLSI indeed performs better than LSI in several text retrieval experiments. The factor representation obtained in PLSI allows us to classify sentences according to the topics they belong to. We will use this ability of PLSI to generate summary of document that are more generic in nature by picking sentences from different topics.

4 Summarization with PLSI

4.1 The Latent Variable Model for Document

Our document model is similar to Aspect Model (Hofmann et al, 1999, Saul and Pereira, 1997) used by Hoffman (1999). The model attempts to associate an unobserved class variable $z \in Z=\{z_1, ..., z_k\}$ (in our case the topics contained in the document), with two sets of observables, documents $(d \in D=\{d_1, ...d_m\}$, sentences in our case) and words $(w \in W=\{w_1, \ldots, w_n\})$ contained in documents. In terms of generative model it can be defined as follows:

-A document d is selected with probability $P(d)$

-A latent class z is selected with probability $P(z|d)$

-A word w is selected with probability $P(w|z)$

For each document-word pair (d,w), the likelihood for each pair can be represented as

$$P(d,w)=P(d)P(w|d)=P(d)\sum_z P(w|z)P(z|d).$$

Following the maximum likelihood principle $P(d)$, $P(z|d)$, $P(w|z)$ are determined by the maximization of of log-likelihood function,

$$L=\sum_d \sum_w n(d,w)\log P(d,w)$$

where $n(d,w)$ denotes the term frequency, i.e., the number of time w occurred in d.

4.2 Maximizing Model Likelihood

Expectation Maximization (EM) is the standard procedure for maximizing likelihood estimation in the presence of latent variables. EM is an iterative procedure and each of the iteration contains two steps. (a) An Expectation (E) step, where the posterior probabilities for latent variable z are computed and (b) Maximization (M) step, where parameters for given posterior probabilities are computed.

The aspect model can be re-parameterized using the Bayes' rule as follows:

$$P(d,w)= \sum_z P(z)\, P(d|z)\, P(w|z)\ .$$

Then using the re-parameterized equation the E-step calculates the posterior for z by

$$P(z \mid d, w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')}$$

This step calculates the probability that word w present in document d can be described by the factor corresponding to z. Subsequently, the M-step re-evaluates the parameters using following equations.

$$P(w \mid z) = \frac{\sum_d n(d,w)P(z|d,w)}{\sum_{d,w'} n(d,w')P(z|d,w')}, \quad (1)$$

$$P(d \mid z) = \frac{\sum_w n(d,w)P(z|d,w)}{\sum_{d',w} n(d',w)P(z|d',w)}, \quad (2)$$

$$P(z) = \frac{\sum_{d,w} n(d,w)P(z|d,w)}{\sum_{d,w} n(d,w)} \quad (3)$$

Alternating the E- and M- steps one approaches a converging point which describes local maximum of the log-likelihood.

We used the tempered EM (TEM) as described by Hofmann (1999). TEM basically introduces a control parameter B, upon which the E-step is modified as,

$$P(z \mid d, w) = \frac{P(z)[P(d|z)P(w|z)]^B}{\sum_{z'} P(z')[P(d|z')P(w|z')]^B} \quad (4)$$

The TEM reduces to original EM if B=1.

4.3 Summarization procedure

We applied PLSI in 4 different ways during the summarization process. We will denote each of the 4 ways as **PROC1**, **PROC2**, **PROC3**, **PROC4**. Each of the four summarization procedure is discussed below.

PROC1 (Dominant topic only): PROC1 consists of the following steps:

a. Each document is represented as term-frequency matrix.
b. *P(w|z), P(d|z), and P(z)* (as in (1), (2), (3)) are calculated until the convergence criteria for EM-algorithm is met. *P(d|z)* represents the importance of document d in given topic represented by *z* and *P(z)* represents the importance of the topic z itself in the document d.
c. *z* with highest probability *P(z)* is picked as the central topic of the document and then the sentences with highest *P(d|z)* score contained in selected topic are picked.
d. The top scoring sentences are used in the summary.

PROC2 (Dominant topic only): PROC2 is the graph based method. PROC2 is similar to PROC1 except for the fact that instead of using term-frequency matrix we use sentence-similarity matrix. Sentence-similarity matrix A is $n \times n$ matrix where n is the number of sentences present in the document. Cosine similarity of each sentence present in the document with respect to all the sentences is calculated. The cosine-similarity values calculated are used instead of term-frequency values as in PROC1. Each entry A_{ij} in matrix A is 0 if the cosine similarity value between sentence i and sentence j is less than threshold value and 1 if greater. We used 0.2 as the threshold value in our experiments after normalizing cosine similarity value. Steps b, c, d from PROC1 are followed after the initial procedure is complete.

This method is analogous to PHITS (Cohn and Chang (2001)) method where the authors utilized PLSI to find communities in hyperlinked environment.

PROC3 (Multiple topics): In both PROC1 and PROC2 we did not take the advantage of the fact that PLSI divides a document into several topics. We only used the sentences from highest ranked topic. In PROC3 we attempt to combine the sentences from different topics while forming the summary. PROC3 can be explained in the following steps.

a. Steps a and b from PROC1 are taken as normal.
b. We mentioned that *P(d|z)* represents the score of the sentence *d* in topic *z*. In this procedure we will create new score R for each sentence using following relation.

$$R=\sum_z P(d|z)P(z)=P(d)$$

Table 1: Evaluation of summaries

The table shows the score of summaries generated using methods described in section 4.3. On the table n means number of topics into which the document has been divided into. Control parameter B from (4) was fixed to 0.75 in this case.

Method Used	**n**	ROUGE-L (recall)	Rouge1	Rouge-2	Rouge-SU4
PROC1	2	0.499	0.557	0.242	0.272
PROC2	2	0.465	0.515	0.227	0.253
PROC3	2	**0.571**	**0.634**	**0.291**	**0.321**
	3	0.571	0.628	0.288	0.318
	4	0.571	0.62	0.28	0.31
	5	0.571	0.613	0.274	0.305
	6	0.5	0.612	0.27	0.302
PROC4	2	0.473	0.508	0.225	0.25
	3	0.472	0.504	0.22	0.245
	4	0.472	0.5	0.219	0.244
	5	0.472	0.492	0.213	0.238
	6	0.471	0.483	0.207	0.231
Compared Methods					
*LexPageRank		0.522	0.577	0.265	0.291
*LSA		0.414	0.463	0.186	0.215
*HITS		0.504	0.562	0.251	0.282

This will essentially score the sentences with generic values or the sentences which have good influence ranging over several topics better.

c. We pick the sentences that score highest score R as the summary.

PROC3 will pick sentences from several topics resulting in better generic summary of the document.

PROC4 (Multiple Topics): PROC4 is essentially PROC3 except for the first few steps. PROC4 does not use the matrix created in PROC1 instead it uses the similarity-matrix produced in PROC2. Once the similarity matrix is created *P(z)* and *P(d|z)* are calculated as in step b of PROC1. Then steps b and c of PROC3 are taken to produce the summary of the document.

5 Experiments and Results

We produced summaries for all the procedures mentioned in section 4.3. We used DUC 2002 data set for summarization. DUC 2002 contains test data for both multiple document and single document summarization. It also contains summaries created by human beings for both single document and multiple document summarization. Our focus in this paper is single document summarization.

After creating summaries we evaluated summaries using ROUGE. ROUGE has been the standard benchmarking technique for summarization tasks adopted by Document Understanding Conference (DUC). We also compared our results with other summarization methods such as LexPageRank (Erkan and Radev, 2004) and Gong and Liu's (2001) LSA-based method. We also compared the results with HITS based method which is similar to LexPageRank but instead of PageRank, HITS is used as ranking algorithm (Klienberg 1998).The results are listed in Table 1.

We used five measures for evaluation, Rouge-L Rouge1, Rouge2, Rouge-SU4 and F_1. These methods are standard methods used in DUC evaluation

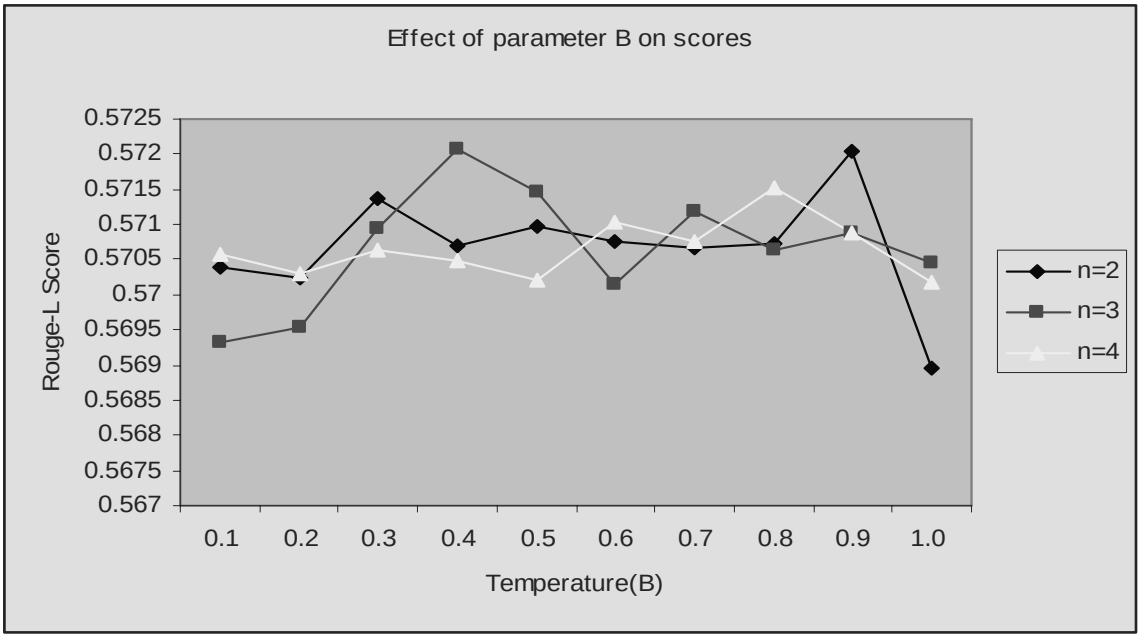


Figure 2: Effect of tempering factor B in the ROUGE-L score for PROC3.

tests and these schemes are known to be very effect tive to calculate the correlation between the summaries. All of the scores can be calculated using Rouge package. Rouge is based on N-gram statistics (Lin and Hovy, 2003). Rouge has been known to highly correlate with human evaluations. According to (Lin and Hovy, 2003), among the methods implemented in ROUGE, ROUGE-N (N=1,2), ROUGE-L, ROUGE-S are relatively simple and work very well even when the length of summary is quite short, which is mostly the case in single document summarization. ROUGE-N, ROUGE-L and ROUGE-S are all basically the recall scores. As DUC keeps the length of the summaries constant recall is the main evaluation criterion. F-measure is also shown in the table as a reference parameter, but since we kept the length of our summaries constant, too, the ROUGE-L, ROUGE-N and ROUGE-S scores carry the highest weight.

As seen on Table 1, the scores gained by PROC1 and PROC2 are less than others. This is mainly because the sentences chosen by these methods were simply chosen from one topic. As PROC3 and PROC4 use sentences from several topics the score of PROC3 and PROC4 were better than PROC1 and PROC2. For methods PROC3 and PROC4 we took the summaries for topics 2 through 6 and found that the method performed well when the number of topics was kept between 2 to 4. But the difference was very small, and in general the performance was quite stable.

We also compared our results to other methods such as LexPageRank and LSA and found that PROC3 performed quite well when compared to those methods. LexPageRank was marginally better in F-measure (F_1) but PROC3 got best recall scores. PROC3 also outperformed LSA by 0.16 in recall (ROUGE-L) scores. Comparison to HITS also shows PROC3 more advantageous.

6 Discussion

In this paper we have argued that choosing sentences from multiple topics makes a better generic summary. It is especially true if we compare our method to graph based ranking methods like HITS and PageRank. Richardson and Domingos (2002) have mentioned that both HITS and PageRank suffer from the topic drift. This not only makes these algorithms susceptible for exclusion of important sentences outside the main topic but miss the sentences from main topic as well. Cohn and Chang (2001) also have shown similar results for HITS. They (Cohn and Chang) have shown that the central topic identified by HITS (principal eigenvector) may not always correspond to the most authoritative topic. The main topic in fact may be represented by smaller eigenvectors rather than the principal one. They also show that the topic segre-

gation in HITS is quite extreme so if we just use principal eigenvector, first there is a chance of being drifted away from the main topic hence producing low quality summary and there is also a chance of missing out other important topics due to the extreme segregation of communities. In PLSI the segregation of topics is not as extreme. If a sentence is related to several topics the sentence can attain high rank in many topics.

We can see from the scores that the performance of graph based algorithms like LexPageRank and HITS are not as good as our method. This can be attributed to the fact that the graph based summarizers take only a central topic under consideration. The method that proved most successful in our summarization was the one where we extracted the sentences that had the most influence in the document.

We used the tempered version of EM-algorithm (4) in our summarization task. We evaluated the effect of tempering factor *B* in performance of summarization for PROC3. We found that that the tempering factor did not influence the results by a big margin. We conducted our experiment using values of **B** from 0.1 through 1.0 incrementing each step by 0.1. The results are shown in Figure 2. In the results shown in Table 1 the value for tempering factor was set to 0.75.

7 Conclusion and Future Work

In this paper we presented a method for creating generic summaries of the documents using PLSI. PLSI allowed us classify the sentences present in the document into several topics. Our summary included sentences from all the topics, which made the generation of generic summary possible. Our experiments showed that the results we obtained in summarization tasks were better than some other methods we compared with. LSA can also be used to summarize documents in similar manner by extracting sentences from several topics, but our experiments showed that PLSI performs better than LSA. In the future we plan to investigate how more recent methods such as LDA (Blei et al) perform in document summarization tasks. We also plan to apply our methods to multiple document summarization.

8 Acknowledgement

We pay our special gratitude to the reviewers who have taken their time to give very useful comments on our work. The comments were very useful for us to as we were able to provide wider perspective on our work with the help of those comments.

References:

Blei D, Ng A, and Jordan M.2003. Journal of Machine Learning Research 3 993-1022.

Brin S and Page L.1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks 30(1-7): 107-117.

Carbonell J and Goldstein J.1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. Proc. ACM SIGIR

Cohn D, Chang H.2001. Learning to probabilistivally identify authoritative documents. Proceedings of 18th International Conference of Machine Learning.

Deerwester S, Dumais ST, Furnas GT, Landauer TK, and Harshman R.1990. Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science.

Erkan G and Radev DR.2004. LexPageRank: Prestige in Multi-Document Text Summarization.EMNLP.

Gong Y and Liu X.2001.Generic text Summarization using relevance measure and latent semantic analysis.Proc ACM SIGIR.

Hofmann, T.1999.Probabilistic Latent Semantic Indexing. Twenty Second International ACM-SIGIR Conference on Informatioln Retrieval.

Hofmann, et al.1998. Unsupervised Learning from Dyadic Data. Technical Report TR-98-042, International Computer Science Insitute, Berkeley, CA.

Kleinberg J.1998. Authoritative sources in a hyperlinked environment. Proc. 9th ACM-SIAM Symposium on Discrete Algorithms.

Mani I. 1999. Advances in Automatic Text Summarization. MIT Press, Cambridge, MA, USA.

Mihalcea R.2005.Language Independent Extractive Summarization. AAAI

Richardson M, Domingos P.2002. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. Advances in Neural Information Processing Systems 14

Saul L and Pereria F.1997.Aggregate and mixed-order Markov models for statistical language processing.Proc 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval.

Identifying Cross-Document Relations between Sentences

Yasunari Miyabe [†§] **Hiroya Takamura** [‡] **Manabu Okumura** [‡]
[†]Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology, Japan
[‡]Precision and Intelligence Laboratory,
Tokyo Institute of Technology, Japan
miyabe@lr.pi.titech.ac.jp, {takamura,oku}@pi.titech.ac.jp

Abstract

A pair of sentences in different newspaper articles on an event can have one of several relations. Of these, we have focused on two, i.e., equivalence and transition. Equivalence is the relation between two sentences that have the same information on an event. Transition is the relation between two sentences that have the same information except for values of numeric attributes. We propose methods of identifying these relations. We first split a dataset consisting of pairs of sentences into clusters according to their similarities, and then construct a classifier for each cluster to identify equivalence relations. We also adopt a "coarse-to-fine" approach. We further propose using the identified equivalence relations to address the task of identifying transition relations.

1 Introduction

A document generally consists of semantic units called sentences and various relations hold between them. The analysis of the structure of a document by identifying the relations between sentences is called *discourse analysis*.

The discourse structure of one document has been the target of the traditional discourse analysis (Marcu, 2000; Marcu and Echihabi, 2002; Yokoyama et al., 2003), based on rhetorical structure theory (RST) (Mann and Thompson, 1987). Inspired by RST, Radev (2000) proposed the cross-document structure theory (CST) for multi-document analysis, such as multi-document summarization, and topic detection and tracking. CST takes the structure of a set of related documents into account. Radev defined relations that hold between sentences across the documents on an event (e.g., an earthquake or a traffic accident).

Radev presented a taxonomy of cross-document relations, consisting of 24 types. In Japanese, Etoh et al. (2005) redefined 14 CST types based on Radev's taxonomy. For example, a pair of sentences with an "equivalence relation" (*EQ*) has the same information on an event. *EQ* can be considered to correspond to the identity and equivalence relations in Radev's taxonomy. A sentence pair with a "transition relation" (*TR*) contains the same numeric attributes with different values. *TR* roughly corresponds to the follow-up and fulfilment relations in Radev's taxonomy. We will provide examples of CST relations:

1. ABC telephone company announced on the 9th that the number of users of its mobile-phone service had reached one million. Users can access the Internet, reserve train tickets, as well as make phone calls through this service.

2. ABC said on the 18th that the number of users of its mobile-phone service had reached 1,500,000. This service includes Internet access, and enables train-ticket reservations and telephone calls.

The pair of the first sentence in 1 and the first sentence in 2 is in *TR*, because the number of users

[§] Yasunari Miyabe currently works at Toshiba Solutions Corporation.

has changed from one million to 1.5 millions, while other things remain unchanged. The pair of the second sentence in 1 and the second sentence in 2 is in *EQ*, because these two sentences have the same information.

Identification of CST relations has attracted more attention since the study of multi-document discourse emerged. Identified CST types are helpful in various applications such as multi-document summarization and information extraction. For example, *EQ* is useful for detecting and eliminating redundant information in multi-document summarization. *TR* can be used to visualize time-series trends.

We focus on the two relations *EQ* and *TR* in the Japanese CST taxonomy, and present methods for their identification. For the identification of *EQ* pairs, we first split a dataset consisting of sentence pairs into clusters according to their similarities, and then construct a classifier for each cluster. In addition, we adopt a coarse-to-fine approach, in which a more general (coarse) class is first identified before the target fine class (*EQ*). For the identification of *TR* pairs, we use *variable noun phrases (VNPs)*, which are defined as noun phrases representing a variable with a number as its value (e.g., stock prices, and population).

2 Related Work

Hatzivassiloglou et al. (1999; 2001) proposed a method based on supervised machine learning to identify whether two paragraphs contain similar information. However, we found it was difficult to accurately identify *EQ* pairs between two sentences simply by using similarities as features. Zhang et al. (2003) presented a method of classifying CST relations between sentence pairs. However, their method used the same features for every type of CST, resulting in low recall and precision. We thus select better features for each CST type, and for each cluster of *EQ*.

The *EQ* identification task is apparently related to Textual Entailment task (Dagan et al., 2005). Entailment is asymmetrical while *EQ* is symmetrical, in the sense that if a sentence entails and is entailed by another sentence, then this sentence pair is in *EQ*. However in the *EQ* identification, we usually need to find *EQ* pairs from an extremely biased dataset of sentence pairs, most of which have no relation at all.

3 Identification of *EQ* pairs

This section explains a method of identifying *EQ* pairs. We regarded the identification of a CST relation as a standard binary classification task. Given a pair of sentences that are from two different but related documents, we determine whether the pair is in *EQ* or not. We use Support Vector Machines (SVMs) (Vapnik, 1998) as a supervised classifier. Please note that one instance consists of a pair of two sentences. Therefore, a similarity value between two sentences is only given to one instance, not two.

3.1 Clusterwise Classification

Although some pairs in *EQ* have quite high similarity values, others do not. Simultaneously using both of these two types of pairs for training will adversely affect the accuracy of classification. Therefore, we propose splitting the dataset first according to similarities of pairs, and then constructing a classifier for each cluster (sub-dataset). We call this method *clusterwise classification*.

We use the following similarity in the cosine measure between two sentences (s_1, s_2):

$$\cos(s_1, s_2) = u_1 \cdot u_2 / |u_1||u_2|, \quad (1)$$

where u_1 and u_2 denote the frequency vectors of content words (nouns, verbs, adjectives) for respective s_1 and s_2. The distribution of the sentence pairs according to the cosine measure is summarized in Table 1. From the table, we can see a large difference in distributions of *EQ* and no-relation pairs. This difference suggests that the clusterwise classification approach is reasonable.

We split the dataset into three clusters: *high-similarity cluster*, *intermediate-similarity cluster*, and *low-similarity cluster*. Intuitively, we expected that a pair in the high-similarity cluster would have many common bigrams, that a pair in the intermediate-similarity cluster would have many common unigrams but few common bigrams, and that a pair in the low-similarity cluster would have few common unigrams or bigrams.

3.2 Two-Stage Identification Method

The number of sentence pairs in *EQ* in the intermediate- or low-similarity clusters is much

Table 1: The distribution of sentence pairs according to the cosine measure (*NO* indicates pairs with no relation. The pairs with other relations are not on the table due to the space limitation)

cos	(0.0, 0.1]	(0.1, 0.2]	(0.2, 0.3]	(0.3, 0.4]	(0.4, 0.5]	(0.5, 0.6]	(0.6, 0.7]	(0.7, 0.8]	(0.8, 0.9]	(0.9, 1.0]
EQ	12	13	21	25	37	61	73	61	69	426
summary	5	5	25	19	22	13	16	6	6	0
refinement	3	4	15	11	12	15	6	6	3	2
NO	194938	162221	68283	28152	11306	4214	1379	460	178	455

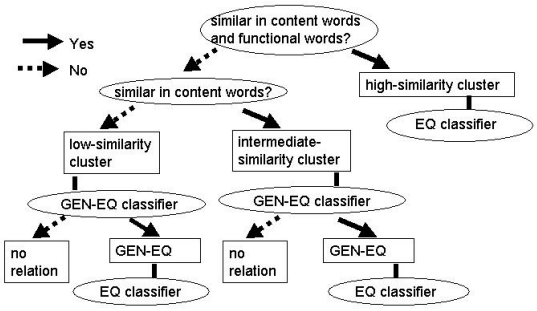


Figure 1: Method of identifying *EQ* pairs

smaller than the total number of sentence pairs as shown in Table 1. These two clusters also contain many pairs that belong to a "summary" and a "refinement" relation, which are very much akin to *EQ*. This may cause difficulties in identifying *EQ* pairs.

We gave a generic name, *GEN(general)-EQ*, to the union of *EQ*, "summary", and "refinement" relations. For pairs in the intermediate- or low-similarity clusters, we propose a two-stage method using *GEN-EQ* on the basis of the above observations, which first identifies *GEN-EQ* pairs between sentences, and then identifies *EQ* pairs from *GEN-EQ* pairs.

This two-stage method can be regarded as a coarse-to-fine approach (Vanderburg and Rosenfeld, 1977; Rosenfeld and Vanderbrug, 1977), which first identifies a coarse class and then finds the target fine class. We used the coarse-to-fine approach on top of the clusterwise classification method as in Fig. 1.

There are by far less *EQ* pairs than pairs without relation. This coarse-to-fine approach will reduce this bias, since *GEN-EQ* pairs outnumber *EQ* pairs.

3.3 Features for identifying *EQ* pairs

Instances (i.e., pairs of sentences) are represented as binary vectors. Numeric features ranging from 0.0 to 1.0 are discretized and represented by 10 binary features (e.g., a feature value of 0.65 is transformed into the vector 0000001000). Let us first explain basic features used in all clusters. We will then explain other features that are specific to a cluster.

3.3.1 Basic features

1. Cosine similarity measures: We use unigram, bigram, trigram, *bunsetsu*-chunk[1] similarities at all the sentence levels, and unigram similarities at the paragraph and the document levels. These similarities are calculated by replacing u_1 and u_2 in Eq. (1) with the frequency vectors of each sentence level.

2. Normalized lengths of sentences: Given an instance of sentence pair s_1 and s_2, we can define features $normL(s_1)$ and $normL(s_2)$, which represent (normalized) lengths of sentences, as:

$$normL(s) = len(s)/EventMax(s), \quad (2)$$

where $len(s)$ is the number of characters in s. $EventMax(s)$ is $\max_{s' \in event(s)} len(s')$, where $event(s)$ is the set of sentences in the event that $doc(s)$ describes. $doc(s)$ is the document containing s.

3. Difference in publication dates: This feature depends on the interval between the publication dates of $doc(s_1)$ and $doc(s_2)$ and is defined as:

$$DateDiff(s_1, s_2) = 1 - \frac{|Date(s_1) - Date(s_2)|}{EventSpan(s_1, s_2)}, \quad (3)$$

where $Date(s)$ is the publication date of an article containing s, and $EventSpan(s_1, s_2)$ is the time span of the event, i.e., the difference between the publication dates for the first and the last articles that are on the same event. For example, if $doc(s_1)$ is published on 1/15/99 and $doc(s_2)$ on 1/17/99, and if the time span of the event ranges from 1/1/99 to 1/21/99, then the feature value is 1-2/20 = 0.9.

[1]Bunsetsu-chunks are Japanese phrasal units usually consisting of a pair of a noun phrase and a case marker.

4. Positions of sentences in documents (Edmundson, 1969): This feature is defined as

$$Posit(s) = lenBef(s)/len(doc(s)), \quad (4)$$

where $lenBef(s)$ is the number of characters before s in the document, and $len(doc(s))$ is the total number of characters in $doc(s)$.

5. Semantic similarities: This feature is measured by Eq. (1) with u_1 and u_2 being the frequency vectors of semantic classes of nouns, verbs, and adjectives. We used the semantic classes in a Japanese thesaurus called 'Goi-taikei' (Ikehara et al., 1997).

6. Conjunction (Yokoyama et al., 2003): Each of 55 conjunctions corresponds to one feature. If a conjunction appears at the beginning of the sentence, the feature value is 1, otherwise 0.

7. Expressions at the end of sentences: Yokoyama et al. (2003) created rules that map sentence endings to their functions. Each function corresponds to a feature. If a function appears in the sentence, the value of the feature for the function is 1, otherwise 0. Functions of sentence endings are past, present, assertion, existence, conjecture, interrogation, judgement, possibility, reason, request, description, duty, opinion, continuation, causation, hearsay, and mode.

8. Named entity: This feature represents similarities measured through named entities in the sentences. Its value is measured by Eq. (1) with u_1 and u_2 being the frequency vectors of the named entities. We used the named-entity chunker bar[2]. The types of named entities are ARTIFACT, DATE, ORGANIZATION, MONEY, LOCATION, TIME, PERCENT, and PERSON.

9. Types of named entities with particle: This feature represents the occurrence of types of named entities accompanied by a case marker (particle). We used 11 different case markers.

3.3.2 Additional features to identify fine class

We will next explain additional features used in identifying *EQ* pairs from *GEN-EQ* pairs.

1. Numbers of words (morphemes) and phrases: These features represent the closeness of the numbers of words and bunsetsu-chunks in the two sentences. This feature is defined as:

$$NumW(s_1, s_2) = 1 - \frac{|frqW(s_1) - frqW(s_2)|}{\max(frqW(s_1), frqW(s_2))}, \quad (5)$$

where $frqW(s)$ indicates the number of words in s. Similarly, $NumP(s_1, s_2)$ is obtained by replacing $frqW$ in Eq. (5) with $frqP$, where $frqP(s)$ indicates the number of phrases in s.

2. Head verb: There are three features of this kind. The first indicates whether the two sentences have the same head verb or not. The second indicates whether the two sentences have a semantically similar head verb or not. If the two verbs have the same semantic class in a thesaurus, they are regarded as being semantically similar. The last indicates whether both sentences have a verb or not. The head verbs are extracted using rules proposed by Hatayama (2001).

3. Salient words: This feature indicates whether the salient words of the two sentences are the same or not. We approximate the salient word with the *ga*- or the *wa*-case word that appears first.

4. Numeric expressions and units (Nanba et al., 2005): The first feature indicates whether the two sentences share a numeric expression or not. The second feature is similarly defined for numeric units.

4 Experiments on identifying *EQ* pairs

We used the Text Summarization Challenge (TSC) 2 and 3 corpora (Okumura et al., 2003) and the Workshop on Multimodal Summarization for Trend Information (Must) corpus (Kato et al., 2005). These two corpora contained 115 sets of related news articles (10 documents per set on average) on various events. A document contained 9.9 sentences on average. Etoh et al. (2005) annotated these two corpora with CST types. There were 471,586 pairs of sentences and 798 pairs of these had *EQ*. We conducted the experiments with 10-fold cross-validation (i.e., approximately 425,000 pairs on average, out of which approximately 700 pairs are in *EQ*, are in the training dataset for each fold). The average, maximum, and minimum lengths of the sentences in the whole datset are shown in Table 2. We used precision, recall, and F-measure as evaluation measures. We used a Japanese morphological analyzer ChaSen[3] to

[2] http://chasen.naist.jp/~masayu-a/p/bar/

[3] http://chasen.naist.jp/hiki/Chasen/

Table 2: Average, max, min lengths of the sentences in the dataset

	average	max	min
# of words	33.27	458	1
# of characters	111.22	1107	2

extract parts-of-speech. and a dependency analyzer CaboCha[4] to extract bunsetsu-chunks.

4.1 Estimation of threshold

We split the set of sentence pairs into clusters according to their similarities in identifying *EQ* pairs as explained. We used 10-fold cross validation again *within the training data* (i.e., the approximately 425,000 pairs above are split into a temporary training dataset and a temporary test dataset 10 times) to estimate the threshold to split the set, to select the best feature set, and to determine the degree of the polynomial kernel function and the value for soft-margin parameter C in SVMs. No training instances are used in the estimation of these parameters.

4.1.1 Threshold between high- and intermediate-similarity clusters

We will first explain how to estimate the threshold between high- and intermediate-similarity clusters.

We expected that a pair in high-similarity cluster would have many common bigrams, and that a pair in intermediate-similarity cluster would have many common unigrams but few common bigrams. We therefore assumed that bigram similarity would be ineffective in intermediate-similarity cluster.

We determined the threshold in the following way for each fold of cross-validation. We decreased the threshold by 0.01 from 1.0. We carried out 10-fold cross-validation within the training data, excluding one of the 14 features (6 cosine similarities and other basic features) for each value of the threshold. If the exclusion of a feature type deteriorates both average precision and recall obtained by the cross-validation within the training data, we call it *ineffective*. We set the threshold to the minimum value for which bigram similarity is not ineffective. We obtain a threshold value for each fold of cross-validation. The average value of threshold was 0.87.

[4] http://chasen.naist.jp/~taku/software/cabocha/

Table 3: Ineffective feature types for each threshold

threshold	ineffective features
0.90	particle, bunsetsu-chunk similarity, semantic similarity
0.89	semantic similarity, expression at end of sentences, **bigram similarity**, particle
0.88	**bigram similarity**
0.87	difference in publication dates, similarity between documents, expression at end of sentences, number of tokens, **bigram similarity**, similarity between paragraphs, positions of sentences, particle
0.86	particle, similarity between documents, **bigram similarity**

Table 4: F-measure calculated by cross-validation within the training data for each threshold in "intermediate-similarity cluster"

threshold	precision	recall	F-measure
0.60	49,71	14.95	22.99
0.59	52.92	15.05	23.44
0.58	55.08	16.64	25.56
0.57	**52.81**	**16.93**	**25.64**
0.56	49.15	14.45	22.34
0.55	51.51	14.84	23.04
0.54	51.89	15.21	23.52
0.53	54.59	13.61	21.78

As an example, we show the table of obtained ineffective feature types for one fold of cross-validation (Table 3). The threshold was set to 0.90 in this fold.

4.1.2 Threshold between intermediate- and low-similarity clusters

We will next explain how to estimate the threshold between intermediate- and low-similarity clusters.

There are numerous no-relation pairs in low-similarity pairs. We expected that this imbalance would adversely affect classification. We therefore simply attemted to exclude low-similarity pairs. We decreased the threshold by 0.01 from the threshold between high- and intermediate-similarity clusters. We chose a value that yielded the best average F-measure calculated by the cross-validation within the training data. The average value of the threshold was 0.57. Table 4 is an example of thresholds and F-measures for one fold.

4.2 Results of identifying *EQ* pairs

The results of *EQ* identification are shown in Table 5. We tested the following models:

Bow-cos: This is the simplest baseline we used. We represented sentences with bag-of-words model. Instances with the cosine similarity in Eq. (1) larger than a threshold were classified as *EQ*. The threshold that yielded the best F-measure in the test

Table 5: Results of identifying *EQ* pairs

	precision	recall	F-measure
Bow-cos	87.29	57.35	69.22
basic features			
Clusterwise	81.98	59.40	68.88
Non-Clusterwise	86.10	59.49	70.36
ClusterC2F	94.96	62.27	75.22
with additional features			
Clusterwise	80.93	59.74	68.63
Non-Clusterwise	86.11	60.16	70.84
ClusterC2F	**94.99**	**62.65**	**75.50**

Table 6: Results with basic features

Results for "high-similarity cluster"			
	precision	recall	F-measure
Clusterwise	94.23	96.83	95.51
Non-clusterwise	95.51	96.29	95.90
ClusterC2F	94.23	96.83	95.51
Results for "intermediate-similarity cluster"			
Clusterwise	42.77	23.03	29.94
Non-clusterwise	53.46	25.31	34.36
ClusterC2F	100.00	36.29	53.25

data was chosen.

Non-Clusterwise: This is a supervised method without the clusterwise approach. One classifier was constructed regardless of the similarity of the instance. We used the second degree polynomial kernel. Soft margin parameter C was set to 0.01.

Clusterwise: This is a clusterwise method without the coarse-to-fine approach. The second degree polynomial kernel was used. Soft margin parameter C was set to 0.1 for high-similarity cluster and 0.01 for the other clusters.

ClusterC2F: This is our model, which integrates clusterwise classification with the coarse-to-fine approach (Figure 1).

Table 5 shows that ClusterC2F yielded the best F-measure regardless of presence of additional features. The difference between ClusterC2F and the others was statistically significant in the Wilcoxon signed rank sum test with 5% significance level.

4.3 Results for each cluster

We examined the results for each cluster. The results with basic features are summarized in Table 6 and those with basic features plus additional features are in Table 7. The tables show that there are no significant differences among the models for high-similarity cluster. However, there are significant differences for intermediate-similarity cluster. We thus concluded that the proposed model (ClusterC2F) works especially well in intermediate-similarity cluster.

Table 7: Results with additional features

Results for "high-similarity cluster"			
	precision	recall	F-measure
Clusterwise	94.23	96.83	95.51
Non-clusterwise	95.70	96.76	96.23
ClusterC2F	94.23	96.83	95.51
Results for "intermediate-similarity cluster"			
Clusterwise	39.77	22.93	29.09
Non-clusterwise	55.61	26.81	36.18
ClusterC2F	100.00	38.06	55.13

5 Identification of *TR* pairs

We regarded the identification of the relations between sentences as binary classification, whether a pair of sentences is classified into *TR* or not. We used SVMs (Vapnik, 1998).

The sentence pairs in *TR* have the same numeric attributes with different values, as mentioned in Introduction. Therefore, VNPs will be good clues for the identification.

5.1 Extraction of VNPs

We extract VNPs in the following way.

1. Search for noun phrases that have numeric expressions (we call them *numeric phrases*).
2. Search for the phrases that the numeric phrases depend on (we call them *predicate phrases*).
3. Search for the noun phrases that depend on the predicate phrases.
4. Extract the noun phrases that depend on the noun phrases found in step 3, except for date expressions. Both the extracted noun phrases and the noun phrases found in step 3 were regarded as VNPs.

In the example in Introduction, "one million" and "1,500,000" are numeric phrases, and "had reached" is a predicate phrase. Then, "the number of users of its mobile-phone service" is a VNP.

5.2 Features for identifying *TR* pairs

We used some features used in *EQ* identification: sentence-level uni-, bi-, tirgrams, and bunsetsu-chunk unigrams, normalized lengths of sentences, difference in publication dates, position of sentences in documents, semantic similarities, conjunctions, expressions at the end of sentences, and named entities. In addition, we use the following features.

1. Similarities through VNPs: The cosine similarity of the frequency vectors of nouns in the VNPs in s_1

and s_2 is used. If there are more than one VNP, the largest cosine similarity is chosen.

2. Similarities through bigrams and trigrams in VNPs: These features are defined similarly to the previous feature, but each VNP is represented by the frequency vector of word bi- and trigrams.

3. Similarities of noun phrases in nominative case: Instances in *TR* often have similar subjects. A noun phrase containing a *ga*-, *wa*-, or *mo*-case is regarded as the subject phrase of a sentence. The similarity is calculated by Eq. (1) with the frequency vectors of nouns in the phrase.

4. Changes in value of numeric attributes: This feature is 1 if the values of the numeric phrases in the two sentences are different, otherwise 0.

5. Presence of numerical units: If a numerical unit is present in both sentences, the value of the feature is 1, otherwise 0.

6. Expressions that mean changes in value: Instances in *TR* often contain those expressions, such as 'reduce' and 'increase' (Nanba et al., 2005). We have three features for each of these expressions. The first feature is 1 if both sentences have the expression, otherwise 0. The second is 1 if s_1 has the expression, otherwise 0. The third is 1 if s_2 has the expression, otherwise 0.

7. Predicates: We define one feature for a predicate. The value of this feature is 1 if the predicate appears in the two sentences, otherwise 0.

8. Reporter: This feature represents who is reporting the incident. This feature is represented by the cosine similarity between the frequency vectors of nouns in phrases respectively expressing reporters in s_1 and s_2. The subjects of verbs such as 'report' and 'announce' are regarded as phrases of the reporter.

5.3 Use of *EQ*

A pair of sentences in *TR* often has a high degree of similarity. Such pairs are likely to be confused with pairs in *EQ*. We used the identified *EQ* pairs for the identification of *TR* in order to circumvent this confusion. Pairs classified as *EQ* with our method were excluded from candidates for *TR*.

Table 8: Results of identifying *TR* pairs

	precision	recall	F-measure
Bow-cos	27.44	41.26	32.96
NANBA	19.85	45.96	27.73
WithoutEq	42.41	47.06	44.61
WithEq	43.13	48.51	45.67
WithEqActual	43.06	48.55	45.64

6 Experiments on identifying *TR* pairs

Most experimental settings are the same as in the experiments of *EQ* identification. Sentence pairs without numeric expressions were excluded in advance and 55,547 pairs were left. This exclusion process does not degrade recall at all, because *TR* pairs *by definition* contain numberic expressions.

We used precision, recall and F-measure for evaluation. We employed 10-fold cross validation.

6.1 Results of identifying *TR* pairs

The results of the experiments are summarized in Table 8. We compared four following models with ours. A linear kernel was used in SVMs and soft margin parameter C was set to 1.0 for all models:

Bow-cos (baseline): We calculated the *similarity through VPNs*. If the similarity was larger than a threshold and the two sentences had the same expressions meaning changes in value and had different values, then this pair was classified as *TR*. The threshold was set to 0.7, which yielded the best F-measure in the test data.

NANBA (Nanba et al., 2005): If the unigram cosine similarity between the two sentences was larger than a threshold and the two sentences had expressions meaning changes in value, then this pair was classified as *TR*. The value of the threshold was set to 0.42, which yielded the best F-measure in the test data.

WithEq (Our method): This model uses the identified *EQ* pairs.

WithoutEq: This model uses no information on *EQ*.

WithEqActual: This model uses the actual *EQ* pairs given by oracle.

The results in Table 8 show that bow-cos is better than NANBA in F-measure. This result suggests that focusing on VNPs is more effective than a simple bag-of-words approach.

WithEq and WithEqActual were better than WithoutEq. This suggests that we successfully excluded *EQ* pairs, which are *TR* look-alikes. WithEq and WithEqActual yielded almost the same F-measure. This means that our *EQ* identifier was good enough

to improve the identification of *TR* pairs.

7 Conclusion

We proposed methods for identifying *EQ* and *TR* pairs in different newspaper articles on an event. We empirically demonstrated that the methods work well in this task.

Although we focused on resolving a bias in the dataset, we can expect that the classification performance will improve by making use of methods developed in different but related tasks such as Textual Entailment recognition on top of our method.

References

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 177–190.

Harold Edmundson. 1969. New methods in automatic extracting. *Journal of ACM*, 16(2):246–285.

Junji Etoh and Manabu Okumura. 2005. Making cross-document relationship between sentences corpus. In *Proceedings of the Eleventh Annual Meeting of the Association for Natural Language Processing (in Japanese)*, pages 482–485.

Mamiko Hatayama, Yoshihiro Matsuo, and Satoshi Shirai. 2001. Summarizing newspaper articles using extracted information and functional words. In *6th Natural Language Processing Pacific Rim Symposium (NLPRS2001)*, pages 593–600.

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of the Empirical Methods for Natural Language Processing*, pages 203–212.

Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown. 2001. Simfinder: A flexible clustering tool for summarization. In *Proceedings of the Workshop on Automatic Summarization*, pages 41–49.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Oyama, and Yoshihiko Hayashi. 1997. *Goi-Taikei – A Japanese Lexicon (in Japanese)*. Iwanami Shoten.

Tsuneaki Kato, Mitsunori Matsushita, and Noriko Kando. 2005. Must:a workshop on multimodal summarization for trend information. In *Proceedings of the NTCIR-5 Workshop Meeting*, pages 556–563.

William Mann and Sandra Thompson. 1987. Rhetorical structure theory: Description and construction of text structures. In Gerard Kempen, editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, pages 85–96. Nijhoff, Dordrecht.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375.

Daniel Marcu. 2000. The rhetorical parsing of unrestricted texts a surface-based approach. *Computational Linguistics*, 26(3):395–448.

Hidetsugu Nanba, Yoshinobu Kunimasa, Shiho Fukushima, Teruaki Aizawa, and Manabu Okumura. 2005. Extraction and visualization of trend information based on the cross-document structure. In *Information Processing Society of Japan, Special Interest Group on Natural Language Processing (IPSJ-SIGNL), NL-168 (in Japanese)*, pages 67–74.

Manabu Okumura, Takahiro Fukushima, and Hidetsugu Nanba. 2003. Text summarization challenge 2 - text summarization evaluation at ntcir workshop 3. In *HLT-NAACL 2003 Workshop: Text Summarization (DUC03)*, pages 49–56.

Dragomir Radev. 2000. A common theory of information fusion from multiple text sources, step one: Cross-document structure. In *Proceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue*, pages 74–83.

Azriel Rosenfeld and Gorden Vanderbrug. 1977. Coarse-fine template matching. *IEEE transactions Systems, Man, and Cybernetics*, 7:104–107.

Gorden Vanderburg and Azriel Rosenfeld. 1977. Two-stage template matching. *IEEE transactions on computers*, 26(4):384–393.

Vladimir Vapnik. 1998. *Statistical Learning Theory*. John Wiley, New York.

Kenji Yokoyama, Hidetsugu Nanba, and Manabu Okumura. 2003. Discourse analysis using support vector machine. In *Information Processing Society of Japan, Special Interest Group on Natural Language Processing (IPSJ-SIGNL), 2003-NL-155 (in Japanese)*, pages 193–200.

Zhu Zhang, Jahna Otterbacher, and Dragomir R.Radev. 2003. Learning cross-document structural relationships using boosting. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, pages 124–130.

Experiments on Semantic-based Clustering for Cross-document Coreference

Horacio Saggion
Department of Computer Science
University of Sheffield
211 Portobello Street - Sheffield, England, UK, S1 4DP
Tel: +44-114-222-1947
Fax: +44-114-222-1810
saggion@dcs.shef.ac.uk

Abstract

We describe clustering experiments for cross-document coreference for the first Web People Search Evaluation. In our experiments we apply agglomerative clustering to group together documents potentially referring to the same individual. The algorithm is informed by the results of two different summarization strategies and an off-the-shelf named entity recognition component. We present different configurations of the system and show the potential of the applied techniques. We also present an analysis of the impact that semantic information and text summarization have in the clustering process.

1 Introduction

Finding information about people on huge text collections or on-line repositories on the Web is a common activity. In ad-hoc Internet retrieval, a request for documents/pages referring to a person name may return thousand of pages which although containing the name, do not refer to the same individual. Cross-document coreference is the task of deciding if two entity mentions in two sources refer to the same individual. Because person names are highly ambiguous (i.e., names are shared by many individuals), deciding if two documents returned by a search engine such as Google or Yahoo! refer to the same individual is a difficult problem.

Automatic techniques for solving this problem are required not only for better access to information but also in natural language processing applications such as multidocument summarization, question answering, and information extraction. Here, we concentrate on the Web People Search Task (Artiles et al., 2007) as defined in the SemEval 2007 Workshop: a search engine user types in a person name as a query. Instead of ranking web pages, an ideal system should organise search results in as many clusters as there are different people sharing the same name in the documents returned by the search engine. The input is, therefore, the results given by a web search engine using a person name as query. The output is a number of sets, each containing documents referring to the same individual. The task is related to the coreference resolution problem disregarding however the linking of mentions of the target entity inside each single document.

Similarly to (Bagga and Baldwin, 1998; Phan et al., 2006), we have addressed the task as a document clustering problem. We have implemented our own clustering algorithms but rely on available extraction and summarization technology to produce document representations used as input for the clustering procedure. We will shown that our techniques produce not only very good results but are also very competitive when compared with SemEval 2007 systems. We will also show that carefully selection of document representation is of paramount importance to achieve good performance. Our system has a similar level of performance as the best system in the recent SemEval 2007 evaluation framework. This paper extends our previous work on this task (Saggion, 2007).

2 Evaluation Framework

The SemEval evaluation has prepared two sets of data to investigate the cross-document coreference problem: one for development and one for testing. The data consists of around 100 Web files per person name, which have been frozen and so, can be used as an static corpus. Each file in the corpus is associated with an integer number which indicates the rank at which the particular page was retrieved by the search engine. In addition to the files themselves, the following information was available: the page title, the url, and the snippet. In addition to the data itself, human assessments are provided which are used for evaluating the output of the automatic systems. The assessment for each person name is a file which contains a number of sets where each set is assumed to contain all (and only those) pages that refer to one individual. The development data is a selection of person names from different sources such as participants of the European Conference on Digital Libraries (ECDL) 2006 and the on-line encyclopædia Wikipedia.

The test data to be used by the systems consisted of 30 person names from different sources: (i) 10 names were selected from Wikipedia; (ii) 10 names were selected from participants in the ACL 2006 conference; and finally, (iii) 10 further names were selected from the US Census. One hundred documents were retrieved using the person name as a query using the search engine Yahoo!.

Metrics used to measure the performance of automatic systems against the human output were borrowed from the clustering literature (Hotho et al., 2003) and they are defined as follows:

$$\text{Precision}(A, B) = \frac{|A \cap B|}{|A|}$$

$$\text{Purity}(C, L) = \sum_{i=1}^{n} \frac{|C_i|}{n} max_j \text{Precision}(C_i, L_j)$$

$$\text{Inverse\_Purity}(C, L) = \sum_{i=1}^{n} \frac{|L_i|}{n} max_j \text{Precision}(L_i, C_j)$$

$$\text{F-Score}_\alpha(C, L) = \frac{\text{Purity}(C, L) * \text{Inverse\_Purity}(C, L)}{\alpha \text{Purity}(C, L) + (1 - \alpha)\text{Inverse\_Purity}(C, L)}$$

where C is the set of clusters to be evaluated and L is the set of clusters produced by the human. Note that purity is a kind of precision metric which rewards a partition which has less noise. Inverse purity is a kind of recall metric. α was set to 0.5 in the SemEval 2007 evaluation. Two simple baseline systems were defined in order to measure if the techniques used by participants were able to improve over them. The all-in-one baseline produces one single cluster – all documents belonging to that cluster. The one-in-one baseline produces n cluster with one different document in each cluster.

3 Agglomerative Clustering Algorithm

Clustering is an important technique used in areas such as information retrieval, text mining, and data mining (Cutting et al., 1992). Clustering algorithms combine data points into groups such that: (i) data points in the same group are similar to each other; and (ii) data points in one group are "different" from data points in a different group or cluster. In information retrieval it is assumed that documents that are similar to each other are likely to be relevant for the same query, and therefore having the document collection organised in clusters can provide improved document access (van Rijsbergen, 1979). Different clustering techniques exist (Willett, 1988) the simplest one being the one-pass clustering algorithm (Rasmussen and Willett, 1987). We have implemented an agglomerative clustering algorithm which is relatively simple, has reasonable complexity, and gave us rather good results. Our algorithm operates in an exclusive way, meaning that a document belongs to one and only one cluster – while this is our working hypothesis, it might not be valid in some cases.

The input to the algorithm is a set of document representations implemented as vectors of terms and weights. Initially, there are as many clusters as input documents; as the algorithm proceeds clusters are merged until a certain termination condition is reached. The algorithm computes the similarity between vector representations in order to decide whether or not to merge two clusters.

The similarity metric we use is the cosine of the angle between two vectors. This metric gives value one for identical vectors and zero for vectors which are orthogonal (non related). Various options have been implemented in order to measure how close

two clusters are, but for the experiments reported here we have used the following approach: the similarity between two clusters (sim_C) is equivalent to the "document" similarity (sim_D) between the two more similar documents in the two clusters – this is known as single linkage in the clustering literature; the following formula is used:

$$\text{sim}_C(\text{C}_1,\text{C}_2) = \max_{d_i \in C_1; d_j \in C_2} \text{sim}_D(\text{d}_i,\text{d}_j)$$

Where C_k are clusters, d_l are document representations (e.g., vectors), and sim_D is the cosine metric given by the following formula:

$$\text{cosine}(d_1, d_2) = \frac{\sum_{i=1}^{n} w_{i,d_1} * w_{i,d_2}}{\sqrt{\sum_{i=1}^{n} (w_{i,d_1})^2} * \sqrt{\sum_{i=1}^{n} (w_{i,d_2})^2}}$$

where $w_{i,d}$ is the weight of term i in document d and n is the numbers of terms.

If this similarity is greater than a threshold – experimentally obtained – the two clusters are merged together. At each iteration the most similar pair of clusters is merged. If this similarity is less than a certain threshold the algorithm stops. Merging two clusters consist of a simple step of *set union*, so there is no re-computation involved – such as computing a cluster centroid.

We estimated the threshold for the clustering algorithm using the ECDL subset of the training data provided by SemEval. We applied the clustering algorithm where the threshold was set to zero. For each document set, purity, inverse purity, and F-score were computed at each iteration of the algorithm, recording the similarity value of each newly created cluster. The similarity values for the best clustering results (best F-score) were recorded, and the maximum and minimum values discarded. The rest of the values were averaged to obtain an estimate of the optimal threshold. The thresholds used for the experiments reported here are as follows: 0.10 for word vectors and 0.12 for named entity vectors (see Section 5 for vector representations).

4 Natural Language Processing Technology

We rely on available extraction and summarization technology in order to linguistically process the documents for creating document representations for clustering. Although the SemEval corpus contains information other than the retrieved pages themselves, we have made no attempt to analyse or use contextual information given with the input document.

Two tools are used: the GATE system (Cunningham et al., 2002) and a summarization toolkit (Saggion, 2002; Saggion and Gaizauskas, 2004) which is compatible with GATE. The input for analysis is a set of documents and a person name (first name and last name). The documents are analysed by the default GATE[1] ANNIE system which creates different types of named entity annotations. No adaptation of the system was carried out because we wanted to verify how far we could go using available tools. Summarization technology was used from single document summarization modules from our summarization toolkit.

The core of the toolkit is a set of summarization modules which compute numeric features for each sentence in the input document, the value of the feature indicates how relevant the information in the sentence is for the feature. The computed values, which are normalised yielding numbers in the interval [0..1] – are combined in a linear formula to obtain a score for each sentence which is used as the basis for sentence selection. Sentences are ranked based on their score and top ranked sentences selected to produce an extract. Many features implemented in this tool have been suggested in past research as valuable for the task of identifying sentences for creating summaries. In this work, summaries are created following two different approaches as described below.

The text and linguistic processors used in our system are: document tokenisation to identify different kinds of words; sentence splitting to segment the text into units used by the summariser; parts-of-speech tagging used for named entity recognition; named entity recognition using a gazetteer lookup module and regular expressions grammars; and named entity coreference module using a rule-based orthographic name matcher to identify name mentions considered equivalent (e.g., "John Smith" and "Mr. Smith"). Named entities of type *Person*, *Organization*, *Address*, *Date*, and *Location* are considered relevant

[1] `http://gate.ac.uk`

document terms and stored in a special named entity called *Mention* as an annotation. The performance of the named entity recogniser on Web data (business news from the Web) is around 0.90 F-score (Maynard et al., 2003).

Coreference chains are created and analysed and if they contain an entity matching the target person's surname, all elements of the chain are marked as a feature of the annotation.

We have tested two summarization conditions in this work: In one set of experiments a sentence belongs to a summary if it contains a mention which is coreferent with the target entity. In a second set of experiments a sentence belongs to a summary if it contains a "biographical pattern". We rely on a number of patterns that have been proposed in the past to identify *descriptive phrases* in text collections (Joho and Sanderson, 2000). The patterns used in the experiments described here are shown in Table 1. In the patterns, *dp* is a *descriptive phrase* that in (Joho and Sanderson, 2000) is taken as a noun phrase. These patterns are likely to capture information which is relevant to create person profiles, as used in DUC 2004 and in TREC QA – to answer definitional questions.

These patterns are implemented as regular expressions using the JAPE language (Cunningham et al., 2002). Our implementation of the patterns make use of coreference information so that *target* is *any* name in text which is coreferent with sought person. In order to implement the *dp* element in the patterns we use the information provided by a noun phrase chunker. The following is one of the JAPE rules for identifying key phrases as implemented in our system:

```
({TargetPerson}
({ Token.string == "is" } |
{Token.string == "was" })
{NounChunk}):annotate --> :annotate.KeyPhrase = {}
```

where *TargetPerson* is the sought entity, and *NounChunk* is a noun chunk. The rule states that when the pattern is found, a *KeyPhrase* should be created.

Some examples of these patterns in text are shown in Table 4. A profile-based summarization system which uses these patterns to create person profiles is reported in (Saggion and Gaizauskas, 2005).

Patterns
target (is | was |...) (a | an | the) dp
target, (who | whose | ...)
target, (a | the | one ...) dp
target, dp
target's
target and others

Table 1: Set of patterns for identifying profile information.

Dickson's invention, the Kinetoscope, was simple: a strip of several images was passed in front of an illuminated lens and behind a spinning wheel.

James Hamilton, 1st earl of Arran

James Davidson, MD, Sports Medicine Orthopedic Surgeon, Phoenix Arizona

As adjutant general, **Davidson was chief** of the State Police, qv which he organized quickly.

Table 2: Descriptive phrases in test documents for different target names.

4.1 Frequency Information

Using language resources creation modules from the summarization tool, two frequency tables are created for each document set (or person) on-the-fly: (i) an inverted document frequency table for *words* (no normalisation is applied); and (ii) an inverted frequency table for *Mentions* (the full entity string is used, no normalisation is applied).

Statistics (term frequencies (tf(Term)) and inverted document frequencies (idf(Term))) are computed over tokens and *Mentions* using tools from the summarization toolkit (see examples in Table 3).

word frequencies	Mention frequencies
of (92)	Jerry Hobbs (80)
Hobbs (92)	Hobbs (56)
Jerry (90)	Krystal Tobias (38)
to (89)	Texas (37)
in (87)	Jerry (36)
and (86)	Laura Hobbs (35)
the (85)	Monday (34)
a (85)	1990 (31)

Table 3: Examples of top frequent terms (words and named entities) and their frequencies in the Jerry Hobbs set.

Using these tables vector representations are created for each document (same as in (Bagga and

Baldwin, 1998)). We use the following formula to compute term weight (N is the number of documents in the input set):

$$\text{weight(Term)} = \text{tf(Term)} * log_2(\frac{N}{\text{idf(Term)}})$$

These vectors are also stored in the GATE documents. Two types of representations were considered for these experiments: (i) full document or summary (terms in the summary are considered for vector creation); and (ii) words are used as terms or *Mentions* are used as terms.

5 Cross-document Coreference Systems

In this section we present results of six different configurations of the clustering algorithm. The configurations are composed of two parts one which indicates where the terms are extracted from and the second part indicates what type of terms were used. The text conditions are as follows: *Full Document* (FD) condition means that the whole document was used for extracting terms for vector creation; *Person Summary* (PS) means that sentences containing the target person name were used to extract terms for vector creation; *Descriptive Phrase* (DP) means that sentences containing a descriptive patterns were used to extract terms for vector creation. The term conditions are: *Words* (W) words were used as terms and *Mentions* (M) named entities were used as terms. Local inverted term frequencies were used to weight the terms.

6 SemEval 2007 Web People Search Results

The best system in SemEval 2007 obtained an F-score of 0.78, the average F-score of all 16 participant systems is 0.60. Baseline *one-in-one* has an F-score of 0.61 and baseline *all-in-one* an F-score of 0.40. Results for our system configurations are presented in Table 4. Our best configuration (FD+W) obtains an F-score of 0.74 (or a fourth position in the SemEval ranking). All our configurations obtained F-scores greater than the average of 0.60 of all participant systems. They also perform better than the two baselines.

Our optimal configurations (FD+W and PS+W) both perform similarly with respect to F-score. While the full document condition favours "inverse purity", summary condition favours "purity". As one may expect, the use of descriptive phrases to create summaries has the effect of increasing purity to one extreme, these expressions are far too restrictive to capture all necessary information for disambiguation.

Configuration	Purity	Inv.Purity	F-Score
FD+W	0.68	0.85	0.74
FD+M	0.62	0.85	0.68
PS+W	0.84	0.70	0.74
PS+M	0.65	0.75	0.64
DP+W	0.90	0.62	0.71
DP+M	0.97	0.53	0.66

Table 4: Results for different clustering configurations. These results are those obtained on the whole set of 30 person names.

7 Semantic-based Experiments

While these results are rather encouraging, they were not optimal. In particular, we were surprised that semantic information performed worst than a simple word-based approach. We decided to investigate whether some types of semantic information might be more helpful than others in the clustering process. We therefore created one vector for each type of information: *Organization*, *Person*, *Location*, *Date*, *Address* in each document and reclustered all test data using one type at a time, without modifying any of the system parameters (e.g., without re-training). The results were very encouraging.

7.1 Results

Results of semantic-based clustering per information type are presented in Tables 5 and 6. Each row

Semantic Type	Purity	Inv.Purity	F-Score	+/-
Organization	0.90	0.72	0.78	+0.10
Person	0.81	0.72	0.75	+0.07
Address	0.82	0.64	0.69	+0.01
Date	0.58	0.85	0.67	-0.01
Location	0.55	0.85	0.64	-0.04

Table 5: Results for full document condition and different semantic information types. Improvements over FD+M are reported.

Semantic Type	Purity	Inv.Purity	F-Score	+/-
Person	0.85	0.64	0.70	+0.06
Organization	0.97	0.57	0.69	+0.05
Date	0.87	0.60	0.68	+0.04
Location	0.82	0.63	0.67	+0.03
Address	0.93	0.54	0.65	+0.01

Table 6: Results for summary condition and different semantic information types. Improvements over PS+M are reported.

in the tables reports results for clustering using one type of information alone. Table 5 reports results for semantic information with full text condition and it is therefore compared to our configuration FD+M which also uses full text condition together with semantic information. The last column in the table shows improvements over that configuration. Using *Organization* type of information in full text condition, not only outperforms the previous system by ten points, also exceeds by a fraction of a point the best system in SemEval 2007 (one point if we consider macro averaged F-score). Statistical tests (*t*-test) show that improvement over FD+M is statistically significant. Other semantic types of information also have improved performance, not all of them however. *Location* and *Date* in the full documents are probably too ambiguous to help disambiguating the target named entity.

Table 6 reports results for semantic information with summary text condition (only personal summaries were tried, experiments using descriptive phrases are underway) and it is therefore compared to our configuration PS+M which also uses summary condition together with semantic information. The last column in the table shows improvements over that configuration. Here all semantic types of information taken individually outperform a system which uses the combination of all types. This is probably because all types of information in a personal summary are somehow related to the target person.

7.2 Results per Person Set

Following (Popescu and Magnini, 2007), we present purity, inverse purity, and F-score results for all our configurations per category (ACL, US Census, Wikipedia) in the test set.

In Tables 7, 8, and 9, results are reported for full

Configuration	Set	Purity	I.Purity	F-Score
FD+Address	ACL	0.86	0.48	0.57
FD+Address	US C.	0.81	0.71	0.75
FD+Address	Wikip.	0.78	0.70	0.73
PS+Address	ACL	0.96	0.38	0.50
PS+Address	US C.	0.94	0.61	0.72
PS+Address	Wikip.	0.88	0.62	0.71
FD+Date	ACL	0.63	0.82	0.69
FD+Date	US C.	0.52	0.87	0.64
FD+Date	Wikip.	0.59	0.85	0.68
PS+Date	ACL	0.88	0.49	0.59
PS+Date	US C.	0.88	0.64	0.72
PS+Date	Wikip.	0.84	0.67	0.72
FD+Location	ACL	0.63	0.78	0.65
FD+Location	US C.	0.52	0.86	0.64
FD+Location	Wikip.	0.49	0.91	0.62
PS+Location	ACL	0.87	0.47	0.54
PS+Location	US C.	0.85	0.66	0.73
PS+Location	Wikip.	0.74	0.75	0.72

Table 7: Results for clustering configurations per person type set (ACL, US Census, and Wikipedia) - Part I.

Configuration	Set	Purity	I.Purity	F-Score
FD+Org.	ACL	0.92	0.57	0.69
FD+Org.	US C.	0.87	0.78	0.82
FD+Org.	Wikip.	0.88	0.79	0.83
PS+Org.	ACL	0.98	0.42	0.54
PS+Org.	US C.	0.95	0.63	0.74
PS+Org.	Wikip.	0.96	0.65	0.77
FD+Person	ACL	0.82	0.66	0.72
FD+Person	US C.	0.81	0.74	0.76
FD+Person	Wikip.	0.77	0.75	0.75
PS+Person	ACL	0.86	0.53	0.63
PS+Person	US C.	0.85	0.6721	0.73
PS+Person	Wikip.	0.82	0.70	0.73

Table 8: Results for clustering configurations per person type set (ACL, US Census, and Wikipedia) - Part II.

document condition(FD), summary condition (PS), word-based representation (W), mention representation (M) – i.e. all types of named entities, and five different mention types: Person, Location, Organization, Date, and Address.

While the Organization type of entity worked better overall, it is not optimal across different categories of people. Note for example that very good results are obtained for the Wikipedia and US Census sets, but rather poor results for the ACL set, where a technique which relies on using full documents and words for document representations works better. These results show that more work is

Configuration	Set	Purity	I.Purity	F-Score
FD+W	ACL	0.73	0.84	0.77
FD+W	US C.	0.54	0.91	0.67
FD+W	Wikip.	0.57	0.91	0.68
FD+M	ACL	0.73	0.76	0.70
FD+M	US C.	0.68	0.82	0.71
FD+M	Wikip.	0.60	0.86	0.68
PS+W	ACL	0.84	0.59	0.65
PS+W	US C.	0.80	0.74	0.75
PS+W	Wikip.	0.70	0.81	0.73
PS+M	ACL	0.75	0.62	0.60
PS+M	US C.	0.71	0.74	0.69
PS+M	Wikip.	0.58	0.83	0.66

Table 9: Results for clustering configurations per person type set (ACL, US Census, and Wikipedia) - Part III.

needed before reaching any conclusions on the best document representation for our algorithm in this task.

8 Related Work

The problem of cross-document coreference has been studied for a number of years now. Bagga and Baldwin (Bagga and Baldwin, 1998) used the vector space model together with summarization techniques to tackle the cross-document coreference problem. Their approach uses vector representations following a bag-of-words approach. Terms for vector representation are obtained from sentences where the target person appears. They have not presented an analysis of the impact of full document versus summary condition and their clustering algorithm is rather under-specified. Here we have presented a clearer picture of the influence of summary vs full document condition in the clustering process.

Mann and Yarowsky (Mann and Yarowsky, 2003) used semantic information extracted from documents referring to the target person in an hierarchical agglomerative clustering algorithm. Semantic information here refers to factual information about a person such as the date of birth, professional career or education. Information is extracted using patterns some of them manually developed and others induced from examples. We differ from this approach in that our semantic information is more general and is not particularly related - although it might be - to the target person.

Phan el al. (Phan et al., 2006) follow Mann and Yarowsky in their use of a kind of biographical information about a person. They use a machine learning algorithm to classify sentences according to particular information types in order to automatically construct a person profile. Instead of comparing biographical information in the person profile altogether as in (Mann and Yarowsky, 2003), they compare each type of information independently of each other, combining them only to make the final decision.

Finally, the best SemEval 2007 Web People Search system (Chen and Martin, 2007) used techniques similar to ours: named entity recognition using off-the-shelf systems. However in addition to semantic information and full document condition they also explore the use of contextual information such as the url where the document comes from. They show that this information is of little help. Our improved system obtained a slightly higher macro-averaged f-score over their system.

9 Conclusions and Future Work

We have presented experiments on cross-document coreference of person names in the context of the first SemEval 2007 Web People Search task. We have designed and implemented a solution which uses an in-house clustering algorithm and available extraction and summarization techniques to produce representations needed by the clustering algorithm. We have presented different approaches and compared them with SemEval evaluation's results. We have also shown that one system which uses one specific type of semantic information achieves state-of-the-art performance. However, more work is needed, in order to understand variation in performance from one data set to another.

Many avenues of improvement are expected. Where extraction technology is concerned, we have used an off-the-shelf system which is probably not the most appropriate for the type of data we are dealing with, and so adaptation is needed here. With respect to the clustering algorithm we plan to carry out further experiments to test the effect of different similarity metrics, different merging criteria including creation of cluster centroids, and cluster distances; with respect to the summarization techniques we intend to investigate how the extraction of sentences

containing pronouns referring to the target entity affects performance, our current version only exploits name coreference. Our future work will also explore how (and if) the use of contextual information available on the web can lead to better performance.

Acknowledgements

We are indebted to the three anonymous reviewers for their extensive suggestions that helped improve this work. This work was partially supported by the EU-funded MUSING project (IST-2004-027097).

References

J. Artiles, J. Gonzalo, and S. Sekine. 2007. The SemEval-2007 WePS Evaluation: Establishing a benchmark for Web People Search Task. In *Proceedings of Semeval 2007, Association for Computational Linguistics*.

A. Bagga and B. Baldwin. 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 79–85.

Y. Chen and J.H. Martin. 2007. Cu-comsem: Exploring rich features for unsupervised web personal named disambiguation. In *Proceedings of SemEval 2007, Assocciation for Computational Linguistics*, pages 125–128.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

Douglass R. Cutting, Jan O. Pedersen, David Karger, and John W. Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329.

A. Hotho, S. Staab, and G. Stumme. 2003. WordNet improves text document clustering. In *Proc. of the SIGIR 2003 Semantic Web Workshop*.

H. Joho and M. Sanderson. 2000. Retrieving Descriptive Phrases from Large Amounts of Free Text. In *Proceedings of Conference on Information and Knoweldge Management (CIKM)*, pages 180–186. ACM.

G. S. Mann and D. Yarowsky. 2003. Unsupervised personal name disambiguation. In W. Daelemans and M. Osborne, editors, *Proceedings of the 7^{th} Conference on Natural Language Learning (CoNLL-2003)*, pages 33–40. Edmonton, Canada, May.

D. Maynard, K. Bontcheva, and H. Cunningham. 2003. Towards a semantic extraction of named entities. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov, editors, *Proceedings of Recent Advances in Natural Language Processing (RANLP'03)*, pages 255–261, Borovets, Bulgaria, Sep. `http://gate.ac.uk/sale/ranlp03/ranlp03.pdf`.

X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. 2006. Personal name resolution crossover documents by a semantics-based approach. *IEICE Trans. Inf. & Syst.*, Feb 2006.

Octavian Popescu and Bernardo Magnini. 2007. Irst-bp: Web people search using name entities. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 195–198, Prague, Czech Republic, June. Association for Computational Linguistics.

E. Rasmussen and P. Willett. 1987. Non-hierarchical document clustering using the icl distribution array processor. In *SIGIR '87: Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 132–139, New York, NY, USA. ACM Press.

H. Saggion and R. Gaizauskas. 2004. Multi-document summarization by cluster/profile relevance and redundancy removal. In *Proceedings of the Document Understanding Conference 2004*. NIST.

H. Saggion and R. Gaizauskas. 2005. Experiments on statistical and pattern-based biographical summarization. In *Proceedings of EPIA 2005*, pages 611–621.

H. Saggion. 2002. Shallow-based Robust Summarization. In *Automatic Summarization: Solutions and Perspectives*, ATALA, December, 14.

H. Saggion. 2007. Shef: Semantic tagging and summarization techniques applied to cross-document coreference. In *Proceedings of SemEval 2007, Assocciation for Computational Linguistics*, pages 292–295.

C.J. van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London.

P. Willett. 1988. Recent trends in hierarchic document clustering: A critical review. *Information Processing & Management*, 24(5):577–597.

Modeling Context in Scenario Template Creation

Long Qiu, Min-Yen Kan, Tat-Seng Chua
Department of Computer Science
National University of Singapore
Singapore, 117590
{qiul,kanmy,chuats}@comp.nus.edu.sg

Abstract

We describe a graph-based approach to Scenario Template Creation, which is the task of creating a representation of multiple related events, such as reports of different hurricane incidents. We argue that context is valuable to identify important, semantically similar text spans from which template slots could be generalized. To leverage context, we represent the input as a set of graphs where predicate-argument tuples are vertices and their contextual relations are edges. A context-sensitive clustering framework is then applied to obtain meaningful tuple clusters by examining their intrinsic and extrinsic similarities. The clustering framework uses Expectation Maximization to guide the clustering process. Experiments show that: 1) our approach generates high quality clusters, and 2) information extracted from the clusters is adequate to build high coverage templates.

1 Introduction

Scenario template creation (STC) is the problem of generating a common semantic representation from a set of input articles. For example, given multiple newswire articles on different hurricane incidents, an STC algorithm creates a template that may include slots for the storm's name, current location, direction of travel and magnitude. Slots in such a scenario template are often to be filled by salient entities in the scenario instance (e.g., "Hurricane Charley" or "the coast area") but some can also be filled by prominent clauses, verbs or adjectives that describe these salient entities. Here, we use the term *salient aspect* (SA) to refer to any of such slot fillers that people would regard as important to describe a particular scenario. Figure 1 shows such a manually-built scenario template in which details about important actions, actors, time and locations are coded as slots.

STC is an important task that has tangible benefits for many downstream applications. In the Message Understanding Conference (MUC), manually-generated STs were provided to guide Information Extraction (IE). An ST can also be viewed as regularizing a set of similar articles as a set of attribute/value tuples, enabling multi-document summarization from filled templates.

Despite these benefits, STC has not received much attention by the community. We believe this is because it is considered a difficult task that requires deep NL understanding of the source articles. A problem in applications requiring semantic similarity is that the same word in different contexts may have different senses and play different roles. Conversely, different words in similar contexts may play similar roles. This problem makes approaches that rely on word similarity alone inadequate.

We propose a new approach to STC that incorporates the use of contextual information to address this challenge. Unlike previous approaches that concentrate on the intrinsic similarity of candidate slot fillers, our approach explicitly models contextual evidence. And unlike approaches to word sense disambiguation (WSD) and other semantic analyses that

use neighboring or syntactically related words as contextual evidence, we define contexts by semantic relatedness which extends beyond sentence boundaries. Figure 2 illustrates a case in point with two excerpts from severe storm reports. Here, although the intrinsic similarity of the main verbs "hit" and "land" is low, their contextual similarity is high as both are followed by clauses sharing similar subjects (hurricanes) and the same verbs. Our approach encodes such contextual information as graphs, mapping the STC problem into a general graph overlay problem that is solvable by a variant of Expectation Maximization (EM).

Our work also contributes resources for STC research. Until now, few scenario templates have been publicly available (as part of MUC), rendering any potential evaluation of automated STC statistically insignificant. As part of our study, we have compiled a set of input articles with annotations that we are making available to the research community.

Scenario Template: Storm

Storm Name	*Charley*
Storm Action	*landed*
Location	*Florida's Gulf coast*
Time	*Friday at 1950GMT*
Speed	*145 mph*
Victim Category 1	*13 people*
Action	*died*
Victim Category 2	*over one million*
Action	*affected*

Figure 1: An example scenario template (filled).

2 Related Work

A natural way to automate the process of STC is to cluster similar text spans in the input article set. SAs then emerge through clustering; if a cluster of text spans is large enough, the aspects contained in it will be considered as SAs. Subsequently, these SAs will be generalized into one or more slots in the template, depending on the definition of the text span. Assuming scenarios are mainly defined by actions, the focus should be on finding appropriate clusters for text spans each of which represents an action. Most of the related work (although they may not directly address STC) shares this assumption and performs

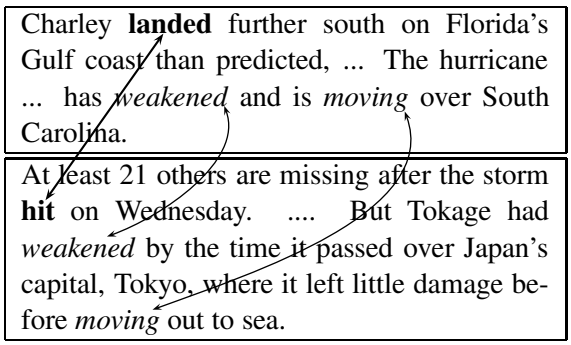


Figure 2: Contextual evidence of similarity. Curved lines indicate similar contexts, providing evidence that "land" and "hit" from two articles are semantically similar.

action clustering accordingly. While the target application varies, most systems that need to group text spans by similarity measures are verb-centric.

In addition to the verb, many systems expand their representation by including named entity tags (Collier, 1998; Yangarber et al., 2000; Sudo et al., 2003; Filatova et al., 2006), as well as restricting matches (using constraints on subtrees (Sudo et al., 2003; Filatova et al., 2006), predicate argument structures (Collier, 1998; Riloff and Schmelzenbach, 1998; Yangarber et al., 2000; Harabagiu and Maiorano, 2002) or semantic roles).

Given these representations, systems then cluster similar text spans. To our knowledge, all current systems use a binary notion of similarity, in which pairs of spans are either similar or not. How they determine similarity is tightly coupled with their text span representation. One criterion used is pattern overlap: for example, (Collier, 1998; Harabagiu and Lacatusu, 2005) judge text spans to be similar if they have similar verbs and share the same verb arguments. Working with tree structures, Sudo et al. and Filatova et al. instead require shared subtrees.

Calculating text span similarity ultimately boils down to calculating word phrase similarity. Approaches such as Yangarber's or Riloff and Schmelzenbach's do not employ a thesaurus and thus are easier to implement, but can suffer from over- or under-generalization. In certain cases, either the same actor is involved in different actions or different verbs realize the same action. Other systems (Collier, 1998; Sudo et al., 2003) do employ

lexical similarity but threshold it to obtain binary judgments. Systems then rank clusters by cluster size and correlation with the relevant article set and equate top clusters as output scenario slots.

3 Context-Sensitive Clustering (CSC)

Automating STC requires handling a larger degree of variations than most previous work we have surveyed. Note that the actors involved in actions in a scenario generally differ from event to event, which makes most related work on text span similarity calculation unsuitable. Also, action participants are not limited to named entities, so our approach needs to process all NPs. As both actions and actors may be realized using different words, a similarity thesaurus is necessary. Our approach to STC uses a thesaurus based on corpus statistics (Lin, 1998) for real-valued similarity calculation. In contrast to previous approaches, we do not threshold word similarity results; we retain their fractional values and incorporate these values holistically. Finally, as the same action can be realized in different constructions, the semantic (not just syntactic) roles of verb arguments must be considered, lest agent and patient roles be confused. For these reasons, we use a semantic role labeler (Pradhan et al., 2004) to provide and delimit the text spans that contain the semantic arguments of a predicate. We term the obtained text spans as *predicate argument tuples* (tuples) throughout the paper. The semantic role labeler reportedly achieves an F_1 measure equal to 68.7% on identification-classification of predicates and core arguments on a newswire text corpus (LDC, 2002). Within the confines of our study, we find it is able to capture most of the tuples of interest.

Our approach explicitly captures contextual evidence. We define a tuple's contexts as other tuples in the same article segment where no topic shift occurs. This definition refines the n-surrounding word constraint commonly used in spelling correction (for example, (Hirst and Budanitsky, 2005)), Word Sense Disambiguation ((Preiss, 2001), (Lee and Ng, 2002), for instance), *etc*. while still ensures the relatedness between a tuple and its contexts. Specifically, a tuple is contextually related to other tuples by two quantifiable contextual relations: *argument-similarity* and *position-similarity*. For our experiments, we use the leads of newswire articles as they normally summarize the news. We also assume a lead qualifies as a single article segment, thus making all of its tuples as potential contexts to each other.

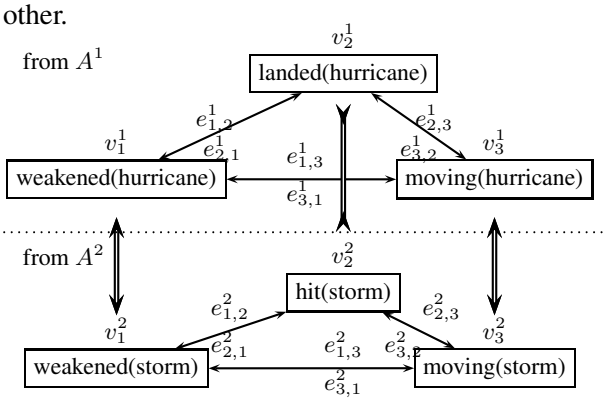


Figure 3: Being similar contexts, "weakened" and "moving" provide contextual evidence that "land" and "hit" are similar.

First, we split the input article leads into sentences and perform semantic role labeling immediately afterwards. Our system could potentially benefit from additional pre-processing such as co-reference resolution. Currently these pre-processing steps have not been properly integrated with the rest of the system, and thus we have not yet measured their impact.

We then transform each lead A^i into a graph $G^i = \{V^i, E^i\}$. As shown in Figure 3, vertices $V^i = \{v_j^i\}(j = 1, ..., N)$ are the N predicate argument tuples extracted from the ith article, and directed edges $E^i = \{e_{m,n}^i = (v_m^i, v_n^i)\}$ reflect contextual relations between tuple v_m^i and v_n^i. Edges only connect tuples from the same article, i.e., within each graph G^i. We differentiate between two types of edges. One is *argument-similarity*, where the two tuples have semantically similar arguments. This models tuple cohesiveness, where the edge weight is determined by the similarity score of the most similar inter-tuple argument pair. The other is *position-similarity*, represented as the offset of the ending tuple with respect to the other, measured in sentences. This edge type is directional to account for simple causality.

Given this set of graphs, the clustering task is to find an optimal alignment of all graphs (i.e., superimposing the set of article graphs to maximize vertex overlap, constrained by the edges). We adapt Expectation Maximization (Dempster et al., 1977) to find

an optimal clustering. This process assigns tuples to suitable clusters where they are semantically similar and share similar contexts with other tuples. Algorithm 1 outlines this alignment process.

Algorithm 1 Graph Alignment($\mathcal{G}$)

```
/*G is a set of graph {G^i}*/
T ← all tuples in G
C ← highly cohesive tuples clusters
other ← remaining tuples semantically connected with C
C[C.length] ← other
repeat
  /*E step*/
  for each i such that i < C.length do
    for each j such that j < C.length do
      if i == j then
        continue;
      re-estimate parameters[C[i], C[j]] /*distribution
      parameters of edges between two clusters*/
  tupleReassigned = false /*reset*/
  /*M step*/
  for each i such that i < T.length do
    aBestLikelihood = T[i].likelihood; /*likelihood of
    being in its current cluster*/
    for each tuple t_contxt that contextually related with
    T[i] do
      for each cluster c_cand, any candidate cluster that
      contextually related with t_contxt.cluster do
        P(T[i] ∈ c_cand) = comb(P_s, P_c)
        likelihood = log(P(T[i] ∈ c_cand))
        if likelihood > aBestLikelihood then
          aBestLikelihood = likelihood
          T[i].cluster = c_cand
          tupleReassigned = true
until tupleReassigned == false /*alignment stable*/
return
```

During initialization, tuples whose pairwise similarity higher than a threshold τ are merged to form highly cohesive seed clusters. To compute a continuous similarity $Sim(t_a, t_b)$ of tuples t_a and t_b, we use the similarity measure described in (Qiu et al., 2006), which linearly combines similarities between the semantic roles shared by the two tuples. Some other tuples are related to these seed clusters by *argument-similarity*. These related tuples are temporarily put into a special "other" cluster. The cluster membership of these related tuples, together with those currently in the seed clusters, are to be further adjusted. The "other" cluster is so called because a tuple will end up being assigned to it if it is not found to be similar to any other tuple. Tuples that are neither similar to nor contextually related by *argument-similarity* to another tuple are termed *singletons* and excluded from being clustered.

We then iteratively (re-)estimate clusters of tuples across the set of article graphs $\mathcal{G}$. In the E-step of the EM algorithm, all contextual relations between each pair of clusters are collected as two set of *edges*. Here we assume *argument-similarity* and *position-similarity* are independent and thus we differentiate them in the computation. Accordingly, there are two sets: $edges_{as}$ and $edges_{ps}$. For simplicity, we assume independent normal distributions for the strength of each set (inter-tuple argument similarity for $edges_{as}$ and sentence distance for $edges_{ps}$). The edge strength distribution parameters for both sets between each pair of clusters are re-estimated based on current edges in $edges_{as}$ and $edges_{ps}$.

In the M-step, we examine each tuple's fitness for belonging to its cluster and relocate some tuples to new clusters to maximize the likelihood given the latest estimated edge strength distributions. In the following equations, we denote the proposition that *predicate argument tuple t_a belongs to cluster c_m* as $t_a \in c_m$; a typical tuple (the centroid) of the cluster c_m as t_{c_m}; and the cluster of t_a as c_{t_a}. The objective function to maximize is:

$$Obj(\mathcal{G}) = \sum_{t_a \in \mathcal{G}} log(P(t_a \in c_{t_a})), \quad (1)$$

$$\text{where } P(t_a \in c_m) = \frac{2P_s(t_a \in c_m)\, P_c(t_a \in c_m)}{P_s(t_a \in c_m) + P_c(t_a \in c_m)}. \quad (2)$$

Equation 2 takes the harmonic mean of two factors: a contextual factor P_c and and a semantic factor P_s:

$$P_c(t_a \in c_m) = \max_{t_b : edges(t_a, t_b) \neq null} \{P(edges(t_a, t_b) | \, edges(c_m, c_{t_b}))\}, \quad (3)$$

$$P_s(t_a \in c_m) = \begin{cases} sim_{default}, & c_m = c_{other}, \\ Sim(t_a, t_{c_m}), & \text{otherwise.} \end{cases} \quad (4)$$

Here the contextual factor P_c models how likely t_a belongs to c_m according to the contextual information, i.e., the conditional probability of the contextual relations between c_m and c_{t_b} given the contextual relations between t_a and one particular context t_b, which maximizes this probability. According to Bayes' theorem, it is computed as shown in Equation 3. In practice, we multiply two conditional probabilities: $P(edge_{as}(t_a, t_b) | edges_{as}(c_m, c_{t_b}))$ and $P(edge_{ps}(t_a, t_b) | edges_{ps}(c_m, c_{t_b}))$, assuming independence between $edges_{as}$ and $edges_{ps}$.

We assume there are still singleton tuples that are not semantically similar to another tuple and should belong to the special "other" cluster. Given that they

are dissimilar to each other, we set $sim_{default}$ to a small nonzero value in Equation 4 to prevent the "other" cluster from expelling them based on their low semantic similarity. Tuples' cluster memberships are recalculated, and the parameters describing the contextual relations between clusters are re-estimated. New EM iterations are performed as long as one or more tuple relocations occur. Once the EM halts, clusters of equivalent tuples are formed. Among these clusters, some correspond to salient actions that, together with their actors, are all SAs to be generalized into template slots. Cluster size is a good indicator of salience, and each large cluster (excluding the "other" cluster) can be viewed as containing instances of a salient action.

Formulating the clustering process as a variant of iterative EM is well-motivated as we consider the similarity scores as noisy and having missing observations. Calculating semantic similarity is at best inaccurate. Thus it is difficult to cluster tuples correctly based only on their semantic similarity. Also to check whether a tuple shares contexts with a cluster of tuples, the cluster has to be relatively clean. An iterative EM as we have proposed naturally improve the cleanness of these tuple clusters gradually as new similarity information comes to light.

4 Evaluation

For STC, we argue that it is crucial to cluster tuples with high recall so that an SA's various surface forms can be captured and the size of clusters can serve as a salience indicator. Meanwhile, precision should not be sacrificed, as more noise will hamper the downstream generalization process which outputs template slots. We conduct experiments designed to answer two relevant research questions:
1) **Cluster Quality:** Whether using contexts (in CSC) produces better clustering results than ignoring it (in the K-means baseline); and
2) **Template Coverage:** Whether slots generalized from CSC clusters cover human-defined templates.

4.1 Data Set and Baseline

A straightforward evaluation of a STC system would compare its output against manually-prepared gold standard templates, such as those found in MUC. Unfortunately, such scenario templates are severely limited and do not provide enough instances for a proper evaluation. To overcome this problem, we have prepared a balanced news corpus, where we have manually selected articles covering 15 scenarios. Each scenario is represented by a total of 45 to 50 articles which describe 10 different events.

Our baseline is a standard K-means clusterer. Its input is identical to that of CSC – the tuples extracted from relevant news articles and are not excluded from being clustered by CSC in the initialization stage (refer to Section 3) – and employs the same tuple similarity measure (Qiu et al., 2006). The differentiating factor between CSC and K-means is the use of contextual evidence. A standard K-means clusterer requires a k to be specified. For each scenario, we set its k as the number of clusters generated by CSC for direct comparison.

We fix the test set for each scenario as ten randomly selected news articles, each reporting a different instance of the scenario; the development set (which also serves as the training set for determining the EM initialization threshold τ and $sim_{default}$ in Equation 4) is a set of ten articles from the "AirlinerCrash" scenario, which are excluded from the test set. Both systems analyze the first 15 sentences of each article, and sentences generate 2 to 3 predicate argument tuples on average, resulting in a total of 10 $\times$ 15 $\times$ (2 to 3) = 300 to 450 tuples for each scenario.

4.2 Cluster Quality

This experiment compares the clustering results of CSC and K-means. We use the standard clustering metrics of *purity* and *inverse purity* (Hotho et al., 2003). The first author manually constructed the gold standard clusters for each scenario using a GUI before conducting any experiments. A special cluster, corresponding to the "other" cluster in the CSC clusters, was created to hold the singleton tuples for each scenario. Table 1 shows this under the column "#Gold Standard Clusters".

Using the manual clusters as the gold standard, we obtain the purity (P) and inverse purity (IP) scores of CSC and K-means on each scenario. In Table 1, we see that CSC outperforms K-means on 10 of 15 scenarios for both P and IP. For the remaining 5 scenarios, where CSC and K-means have comparable

P scores, the IP scores of CSC are all significantly higher than that of K-means. This suggests clusters tend to be split apart more in K-means than in CSC when they have similar purity. One thing worth mentioning here is that the "other" cluster normally is relatively large for each scenario, and thus may skew the results. To remove this effect, we excluded tuples belonging to the CSC "other" cluster from the K-means input, generating one fewer cluster. Running the evaluation again, the resulting P-IP scores again show that CSC outperforms the baseline K-means. We only report the results for all tuples in our paper for simplicity.

Scenario	#Gold Std. Clusters	CSC		K-means	
		P	IP	P	IP
AirlinerCrash	23	**.61**	**.42**	.52	.28
Earthquake	18	**.60**	**.44**	.53	.30
Election	10	**.77**	**.49**	.75	.21
Fire	14	**.65**	**.44**	.64	.26
LaunchEvent	12	**.77**	**.37**	.73	.22
Layoff	10	**.71**	**.28**	.70	.19
LegalCase	8	.75	**.37**	.75	.18
Nobel	6	.77	**.28**	.77	.19
Obituary	7	**.85**	**.46**	.81	.28
RoadAccident	20	**.61**	**.49**	.56	.40
SoccerFinal	5	.88	**.39**	.88	.15
Storm	14	.61	**.31**	.61	.22
Tennis	6	.87	**.19**	.87	.12
TerroristAttack	14	**.64**	**.48**	.62	.25
Volcano	16	**.68**	**.38**	.66	.17
Average	12.2	**.72**	**.39**	.69	.23

Table 1: CSC outperforms K-means with respect to the purity (P) and inverse purity (IP) scores.

A close inspection of the results reveals some problematic cases. One issue worth mentioning is that for certain actions both CSC and K-means produce split clusters. In the CSC case, we traced this problem back to the thesaurus, where predicates for one action seem to belong to two or more totally dissimilar semantic categories. The corresponding tuples are thus assigned to different clusters as their low semantic similarity forces the tuples to remain separate, despite the shared contexts trying to join them. One example is "blast (off)" and "lift (off)" in the "Launch Event" scenario. The thesaurus shows the two verbs are dissimilar and the corresponding tuples end up being in two split clusters. This can not be solved easily without an improved thesaurus. We are considering adding a prior to model the optimal size for clusters, which may help to compact such cases.

4.3 Template Coverage

We also assess how well the resulting, CSC-generated tuple clusters serve in creating good scenario template slots. We start from the top largest clusters from each scenario, and decompose each of them into six sets: the *predicates*, *agents*, *patients*, *predicate modifiers*, *agent modifiers* and *patient modifiers*. For each of the first three sets for each cluster, we create a generalized term to represent it using an extended version of a generalization algorithm (Tseng et al., 2006). These terms are deemed output slots, and are put into the template with their agent-predicate-patient relations preserved. The size of the template may increase when more clusters are generalized, as new slots may result.

We manually compare the slots that are output from the system with those defined in existing scenario templates in MUC. The results here are only indicative and not conclusive, as there are only two MUC7 templates available for comparison: *Aviation Disaster* and *Launch Event*.

Template	semantic role	general term
cluster 1	**action**	crash
	agent	aircraft
	patient	—
cluster 2	**action**	kill
	agent	heavier-than-air-craft
	patient	people

Figure 4: Automated scenario template of "AviationDisaster".

Figure 4 shows an excerpt of the automatically generated template "AviationDisaster" ("AirlinerCrash" in our corpus) where the semantic roles in the top two biggest clusters have been generalized. Their modifiers are quite semantically diverse, as shown in Table 2. Thus, generalization (probably after a categorization operation) remains as a challenging problem.

Nonetheless, the information contained in these semantic roles and their modifiers covers human-

semantic role	modifier head samples
agent:aircraft	A, U.N., The, Swiss, Canadian-built, AN, China, CRJ-200, military, Iranian, Air, refueling, US, ...
action:crash	Siberia, mountain, rain, Tuesday, flight, Sharjah, flames, Sunday, board, Saturday, 225, Rockaway, approach, United, mountain, hillside
patient:people	all, 255, 71

Table 2: Sample automatically detected modifier heads of different semantic roles.

	AviationDisaster
*	AIRCRAFT
*	AIRLINE
	DEPARTURE_POINT
	DEPARTURE_DATE
*	AIRCRAFT_TYPE
*	CRASH_DATE
*	CRASH_SITE
	CAUSE_INFO
*	VICTIMS_NUM

	LaunchEvent
*	VEHICLE
*	VEHICLE_TYPE
*	VEHICLE_OWNER
*	PAYLOAD
	PAYLOAD_TYPE
	PAYLOAD_FUNC
*	PAYLOAD_OWNER
	PAYLOAD_ORIGIN
*	LAUNCH_DATE
*	LAUNCH_SITE
	MISSION_TYPE
	MISSION_FUNCTION
	MISSION_STATUS

Figure 5: MUC-7 template coverage: asterisks marking all the slots that could be automatically generated.

defined scenario templates quite well. The two MUC7 templates are shown as a list of slots in Figure 5, where horizontal lines delimit slots about different semantic roles, and asterisks mark all the slots that could be automatically generated by our system once it has an improved generalizer. We can see substantial amount of overlap, indicating that a STC system powered by CSC is able to capture scenarios' important facts.

5 Conclusion

We have introduced a new context-sensitive approach to the scenario template creation (STC) problem. Our method leverages deep NL processing, using semantic role labeler's structured semantic tuples as input. Despite the use of deeper semantics, we believe that intrinsic semantic similarity by itself is not sufficient for clustering. We have shown this through examples and argue that an approach that considers contextual similarity is necessary. A key aspect of our work is the incorporation of such contextual information. Our approach uses a notion of context that combines two aspects: positional similarity (when two tuples are adjacent in the text), and argument similarity (when they have similar arguments). The set of relevant articles are represented as graphs where contextual evidence is encoded.

By mapping our problem into a graphical formalism, we cast the STC clustering problem as one of multiple graph alignment. Such a graph alignment is solved by an adaptation of EM, which handles contexts and real-valued similarity by treating both as noisy and potentially unreliable observations.

While scenario template creation (STC) is a difficult problem, its evaluation is arguably more difficult due to the dearth of suitable resources. We have compiled and released a corpus of over 700 newswire articles that describe different instances of 15 scenarios, as a suitable input dataset for further STC research. Using this dataset, we have evaluated and analyzed our context-sensitive approach. While our results are indicative, they show that considering contextual evidence improves performance.

Acknowledgments

The authors are grateful to Kathleen R. McKeown and Elena Filatova at Columbia University for their stimulating discussions and comments over different stages of the preparation of this paper.

References

Robin Collier. 1998. *Automatic Template Creation for Information Extraction*. Ph.D. thesis, University of Sheffield, UK.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *JRSSB*, 39:1–38.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. 2006. Automatic creation of domain templates. In *Proceedings of the COLING/ACL '06*.

Sanda M. Harabagiu and V. Finley Lacatusu. 2005. Topic themes for multi-document summarization. In *Proceedings of SIGIR '05*.

Sanda M. Harabagiu and S. J. Maiorano. 2002. Multi-document summarization with GISTEXTER. In *Proceedings of LREC '02*.

Graeme Hirst and Alexander Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1).

Andreas Hotho, Steffen Staab, and Gerd Stumme. 2003. WordNet improves text document clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*.

LDC. 2002. The aquaint corpus of english news text, catalog no. LDC2002t31.

Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of EMNLP '02*.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL '98*.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of HLT/NAACL '04*.

Judita Preiss. 2001. Local versus global context for wsd of nouns. In *Proceedings of CLUK4*.

Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2006. Paraphrase recognition via dissimilarity significance classification. In *Proceedings of EMNLP '06*.

Ellen Riloff and M. Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proceedings of WVLC '98*.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. 2003. An improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of ACL '03*.

Yuen-Hsien Tseng, Chi-Jen Lin, Hsiu-Han Chen, and Yu-I Lin. 2006. Toward generic title generation for clustered documents. In *Proceedings of AIRS '06*.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. 2000. Unsupervised discovery of scenario-level patterns for information extraction. In *Proceedings of ANLP '00*.

Cross Language Text Categorization Using a Bilingual Lexicon

Ke Wu, Xiaolin Wang and **Bao-Liang Lu***
Department of Computer Science and Engineering, Shanghai Jiao Tong University
800 Dong Chuan Rd., Shanghai 200240, China
{wuke,arthur_general,bllu}@sjtu.edu.cn

Abstract

With the popularity of the Internet at a phenomenal rate, an ever-increasing number of documents in languages other than English are available in the Internet. Cross language text categorization has attracted more and more attention for the organization of these heterogeneous document collections. In this paper, we focus on how to conduct effective cross language text categorization. To this end, we propose a cross language naive Bayes algorithm. The preliminary experiments on collected document collections show the effectiveness of the proposed method and verify the feasibility of achieving performance close to monolingual text categorization, using a bilingual lexicon alone. Also, our algorithm is more efficient than our baselines.

1 Introduction

Due to the popularity of the Internet, an ever-increasing number of documents in languages other than English are available in the Internet. The organization of these heterogeneous document collections increases cost of human labor significantly. On the one hand, experts who know different languages are required to organize these collections. On the other hand, maybe there exist a large amount of labelled documents in a language (e.g. English) which are in the same class structure as the unlabelled documents in another language. As a result, how to exploit the existing labelled documents in some language (e.g. English) to classify the unlabelled documents other than the language in multilingual scenario has attracted more and more attention (Bel et al., 2003; Rigutini et al., 2005; Olsson et al., 2005; Fortuna and Shawe-Taylor, 2005; Li and Shawe-Taylor, 2006; Gliozzo and Strapparava, 2006). We refer to this task as cross language text categorization. It aims to extend the existing automated text categorization system from one language to other languages without additional intervention of human experts. Formally, given two document collections $\{\mathcal{D}_e, \mathcal{D}_f\}$ from two different languages e and f respectively, we use the labelled document collection $\mathcal{D}_e$ in the language e to deduce the labels of the document collection $\mathcal{D}_f$ in the language f via an algorithm $\mathcal{A}$ and some external bilingual resources.

Typically, some external bilingual lexical resources, such as machine translation system (MT), large-scale parallel corpora and multilingual ontology etc., are used to alleviate cross language text categorization. However, it is hard to obtain them for many language pairs. In this paper, we focus on using a cheap bilingual resource, e.g. bilingual lexicon without any translation information, to conduct cross language text categorization. To my knowledge, there is little research on using a bilingual lexicon alone for cross language text categorization.

In this paper, we propose a novel approach for cross language text categorization via a bilingual lexicon alone. We call this approach as Cross Language Naive Bayes Classifier (CLNBC). The proposed approach consists of two main stages. The first stage is to acquire a probabilistic bilingual lex-

*Corresponding author.

icon. The second stage is to employ naive Bayes method combined with Expectation Maximization (EM) (Dempster et al., 1977) to conduct cross language text categorization via the probabilistic bilingual lexicon. For the first step, we propose two different methods. One is a naive and direct method, that is, we convert a bilingual lexicon into a probabilistic lexicon by simply assigning equal translation probabilities to all translations of a word. Accordingly, the approach in this case is named as CLNBC-D. The other method is to employ an EM algorithm to deduce the probabilistic lexicon. In this case, the approach is called as CLNBC-EM. Our preliminary experiments on our collected data have shown that the proposed approach (CLNBC) significantly outperforms the baselines in cross language case and is close to the performance of monolingual text categorization.

The remainder of this paper is organized as follows. In Section 2, we introduce the naive Bayes classifier briefly. In Section 3, we present our cross language naive Bayes algorithm. In Section 4, evaluation over our proposed algorithm is performed. Section 5 is conclusions and future work.

2 The Naive Bayes Classifier

The naive Bayes classifier is an effective known algorithm for text categorization (Domingos and Pazzani, 1997). When it is used for text categorization task, each document $d \in \mathcal{D}$ corresponds to an example. The naive Bayes classifier estimates the probability of assigning a class $c \in \mathcal{C}$ to a document d based on the following Bayes' theorem.

$$P(c|d) \propto P(d|c)P(c) \tag{1}$$

Then the naive Bayes classifier makes two assumptions for text categorization. Firstly, each word in a document occurs independently. Secondly, there is no linear ordering of the word occurrences.

Therefore, the naive Bayes classifier can be further formalized as follows:

$$P(c|d) \propto P(c) \prod_{w \in d} P(w|c) \tag{2}$$

The estimates of $P(c)$ and $P(w|c)$ can be referred to (McCallum and Nigam, 1998)

Some extensions to the naive Bayes classifier with EM algorithm have been proposed for various text categorization tasks. The naive Bayes classifier was combined with EM algorithm to learn the class label of the unlabelled documents by maximizing the likelihood of both labelled and unlabelled documents (Nigam et al., 2000). In addition, the similar way was adopted to handle the problem with the positive samples alone (Liu et al., 2002). Recently, transfer learning problem was tackled by applying EM algorithm along with the naive Bayes classifier (Dai et al., 2007). However, they all are monolingual text categorization tasks. In this paper, we apply a similar method to cope with cross language text categorization using bilingual lexicon alone.

3 Cross Language Naive Bayes Classifier Algorithm

In this section, a novel cross language naive Bayes classifier algorithm is presented. The algorithm contains two main steps below. First, generate a probabilistic bilingual lexicon; second, apply an EM-based naive Bayes learning algorithm to deduce the labels of documents in another language via the probabilistic lexicon.

Table 1: Notations and explanations.

Notations	Explanations
e	Language of training set
f	Language of test set
d	Document
$\mathcal{D}_e$	Document collection in language e
$\mathcal{D}_f$	Document collection in language f
$\mathcal{V}_e$	Vocabulary of language e
$\mathcal{V}_f$	Vocabulary of language f
$\mathcal{L}$	Bilingual lexicon
$\mathcal{T} \subseteq \mathcal{V}_e \times \mathcal{V}_f$	Set of links in $\mathcal{L}$
λ_γ	Set of words whose translation is γ in $\mathcal{L}$
$E \subseteq \mathcal{V}_e$	Set of words of language e in $\mathcal{L}$
$w_e \in E$	Word in E
$F \subseteq \mathcal{V}_f$	Set of words of language f in $\mathcal{L}$
$w_f \in F$	Word in F
$\|E\|$	Number of distinct words in set E
$\|F\|$	Number of distinct words in set F
$N(w_e)$	Word frequency in $\mathcal{D}_e$
$N(w_f, d)$	Word frequency in d in language f
$\mathfrak{D}_e$	Data distribution in language e

For ease of description, we first define some notations in Table 1. In the next two sections, we detail the mentioned-above two steps separately.

3.1 Generation of a probabilistic bilingual lexicon

To fill the gap between different languages, there are two different ways. One is to construct the multilingual semantic space, and the other is to transform documents in one language into ones in another language. Since we concentrate on use of a bilingual lexicon, we adopt the latter method. In this paper, we focus on the probabilistic model instead of selecting the best translation. That is, we need to calculate the probability of the occurrence of word w_e in language e given a document d in language f, i.e. $P(w_e|d)$. The estimation can be calculated as follows:

$$P(w_e|d) = \sum_{w_f \in d} P(w_e|w_f, d)P(w_f|d) \quad (3)$$

Ignoring the context information in a document d, the above probability can be approximately estimated as follows:

$$P(w_e|d) \simeq \sum_{w_f \in d} P(w_e|w_f)P(w_f|d) \quad (4)$$

where $P(w_f|d)$ denotes the probability of occurrence of w_f in d, which can be estimated by relative frequency of w_f in d.

In order to induce $P(w_e|d)$, we have to know the estimation of $P(w_e|w_f)$. Typically, we can obtain a probabilistic lexicon from a parallel corpus. In this paper, we concentrate on using a bilingual lexicon alone as our external bilingual resource. Therefore, we propose two different methods for cross language text categorization.

First, a naive and direct method is that we assume a uniform distribution on a word's distribution. Formally, $P(w_e|w_f) = \frac{1}{\lambda_{w_f}}$, where $(w_e, w_f) \in \mathcal{T}$; otherwise $P(w_e|w_f) = 0$.

Second, we can apply EM algorithm to deduce the probabilistic bilingual lexicon via the bilingual lexicon $\mathcal{L}$ and the training document collection at hand. This idea is motivated by the work (Li and Li, 2002).

We can assume that each word w_e in language e is independently generated by a finite mixture model as follows:

$$P(w_e) = \sum_{w_f \in F} P(w_f)P(w_e|w_f) \quad (5)$$

Therefore we can use EM algorithm to estimate the parameters of the model. Specifically speaking, we can iterate the following two step for the purpose above.

- E-step

$$P(w_f|w_e) = \frac{P(w_f)P(w_e|w_f)}{\sum_{w \in F} P(w)P(w_e|w)} \quad (6)$$

- M-step

$$P(w_e|w_f) = \frac{(N(w_e)+1)P(w_f|w_e)}{\sum_{w \in E} (N(w)+1) P(w_f|w)} \quad (7)$$

$$\begin{aligned} P(w_f) &= \lambda \cdot \sum_{w_e \in E} P(w_e)P(w_f|w_e) \\ &+ (1-\lambda) \cdot P^{'}(w_f) \end{aligned} \quad (8)$$

where $0 \leq \lambda \leq 1$, and

$$P^{'}(w_f) = \frac{\sum_{d \in \mathcal{D}_f} N(w_f, d) + 1}{\sum_{w_f \in F} \sum_{d \in \mathcal{D}_f} N(w_f, d) + |F|} \quad (9)$$

The detailed algorithm can be referred to Algorithm 1. Furthermore, the probability that each word in language e occurs in a document d in language f, $P(w_e|d)$, can be calculated according to Equation (4).

3.2 EM-based Naive Bayes Algorithm for Labelling Documents

In this sub-section, we present an EM-based semi-supervised learning method for labelling documents in different language from the language of training document collection. Its basic model is naive Bayes model. This idea is motivated by the transfer learning work (Dai et al., 2007). For simplicity of description, we first formalize the problem. Given the labelled document set $\mathcal{D}_e$ in the source language and the unlabelled document set $\mathcal{D}_f$, the objective is to find the maximum a posteriori hypothesis h_{MAP}

Algorithm 1 EM-based Word Translation Probability Algorithm

Input: Training document collection $\mathcal{D}_e^{(l)}$, bilingual lexicon $\mathcal{L}$ and maximum times of iterations T
Output: Probabilistic bilingual lexicon $P(w_e|w_f)$

1: Initialize $P^{(0)}(w_e|w_f) = \frac{1}{|\lambda_{w_f}|}$, where $(w_e, w_f) \in \mathcal{T}$; otherwise $P^{(0)}(w_e|w_f) = 0$
2: Initialize $P^{(0)}(w_f) = \frac{1}{|F|}$
3: **for** t =1 to T **do**
4: Calculate $P^{(t)}(w_f|w_e)$ based on $P^{(t-1)}(w_e|w_f)$ and $P^{(t-1)}(w_f)$ according to Equation (6)
5: Calculate $P^{(t)}(w_e|w_f)$ and $P^{(t)}(w_f)$ based on $P^{(t)}(w_f|w_e)$ according to Equation (7) and Equation (8)
6: **end for**
7: **return** $P^{(T)}(w_e|w_f)$

from the hypothesis space H under the data distribution of the language e, $\mathfrak{D}_e$, according to the following formula.

$$h_{MAP} = \arg\max_{h \in H} P_{\mathfrak{D}_e}(h|\mathcal{D}_e, \mathcal{D}_f) \quad (10)$$

Instead of trying to maximize $P_{\mathfrak{D}_e}(h|\mathcal{D}_e, \mathcal{D}_f)$ in Equation (10), we can work with $\ell(h|\mathcal{D}_e, \mathcal{D}_f)$, that is, $\log\left(P_{\mathfrak{D}_e}(h)P(\mathcal{D}_e, \mathcal{D}_f|h)\right)$. Then, using Equation (10), we can deduce the following equation.

$$\begin{aligned}\ell(h|\mathcal{D}_e, \mathcal{D}_f) \propto \log P_{\mathfrak{D}_e}(h) \\ + \sum_{d \in \mathcal{D}_e} \log \sum_{c \in \mathcal{C}} P_{\mathfrak{D}_e}(d|c)P_{\mathfrak{D}_e}(c|h) \\ + \sum_{d \in \mathcal{D}_f} \log \sum_{c \in \mathcal{C}} P_{\mathfrak{D}_e}(d|c)P_{\mathfrak{D}_e}(c|h)\end{aligned} \quad (11)$$

EM algorithm is applied to find a local maximum of $\ell(h|\mathcal{D}_e, \mathcal{D}_f)$ by iterating the following two steps:

- E-step:

$$P_{\mathfrak{D}_e}(c|d) \propto P_{\mathfrak{D}_e}(c)P_{\mathfrak{D}_e}(d|c) \quad (12)$$

- M-step:

$$P_{\mathfrak{D}_e}(c) = \sum_{k \in \{e,f\}} P_{\mathfrak{D}_e}(\mathcal{D}_k)P_{\mathfrak{D}_e}(c|\mathcal{D}_k) \quad (13)$$

$$P_{\mathfrak{D}_e}(w_e|c) = \sum_{k \in \{e,f\}} P_{\mathfrak{D}_e}(\mathcal{D}_k)P_{\mathfrak{D}_e}(w_e|c, \mathcal{D}_k) \quad (14)$$

Algorithm 2 Cross Language Naive Bayes Algorithm

Input: Labelled document collection $\mathcal{D}_e$, unlabelled document collection $\mathcal{D}_f$, a bilingual lexicon $\mathcal{L}$ from language e to language f and maximum times of iterations T.
Output: the class label of each document in $\mathcal{D}_f$

1: Generate a probabilistic bilingual lexicon;
2: Calculate $P(w_e|d)$ according to Equation (4).
3: Initialize $P^{(0)}_{\mathfrak{D}_e}(c|d)$ via the traditional naive Bayes model trained from the labelled collection $\mathcal{D}_e^{(l)}$.
4: **for** t =1 to T **do**
5: **for all** $c \in \mathcal{C}$ **do**
6: Calculate $P^{(t)}_{\mathfrak{D}_e}(c)$ based on $P^{(t-1)}_{\mathfrak{D}_e}(c|d)$ according to Equation (13)
7: **end for**
8: **for all** $w_e \in E$ **do**
9: Calculate $P^{(t)}_{\mathfrak{D}_e}(w_e|c)$ based on $P^{(t-1)}_{\mathfrak{D}_e}(c|d)$ and $P(w_e|d)$ according to Equation (14)
10: **end for**
11: **for all** $d \in \mathcal{D}_f$ **do**
12: Calculate $P^{(t)}_{\mathfrak{D}_e}(c|d)$ based on $P^{(t)}_{\mathfrak{D}_e}(c)$ and $P^{(t)}_{\mathfrak{D}_e}(w_e|c)$ according to Equation (12)
13: **end for**
14: **end for**
15: **for all** $d \in \mathcal{D}_f$ **do**
16: $c = \arg\max_{c \in \mathcal{C}} P^{(T)}_{\mathfrak{D}_e}(c|d)$
17: **end for**

For the ease of understanding, we directly put the details of the algorithm in cross-language text categorization algorithmin which we ignore the detail of the generation algorithm of a probabilistic lexicon.

In Equation (12), $P_{\mathfrak{D}_e}(d|c)$ can be calculated by

$$P_{\mathfrak{D}_e}(d|c) = \prod_{\{w_e | w_e \in \lambda_{w_f} \wedge w_f \in d\}} P_{\mathfrak{D}_e}(w_e|c)^{N_{\mathfrak{D}_e}(w_e,d)} \quad (15)$$

where $N_{\mathfrak{D}_e}(w_e, d) = |d|P_{\mathfrak{D}_e}(w_e|d)$.

In Equation (13), $P_{\mathfrak{D}_e}(c|\mathcal{D}_k)$ can be estimated as follows:

$$P_{\mathfrak{D}_e}(c|\mathcal{D}_k) = \sum_{d \in \mathcal{D}_k} P_{\mathfrak{D}_e}(c|d) P_{\mathfrak{D}_e}(d|\mathcal{D}_k) \quad (16)$$

In Equation (14), similar to section 2, we can estimate $P_{\mathfrak{D}_e}(w_e|c, \mathcal{D}_k)$ through Laplacian smoothing as follows:

$$P_{\mathfrak{D}_e}(w_e|c, \mathcal{D}_k) = \frac{1 + N_{\mathfrak{D}_e}(w_e, c, \mathcal{D}_k)}{|\mathcal{V}_k| + N_{\mathfrak{D}_e}(c, \mathcal{D}_k)} \quad (17)$$

where

$$N_{\mathfrak{D}_e}(w_e, c, \mathcal{D}_k) = \sum_{d \in \mathcal{D}_k} |d| P_{\mathfrak{D}_e}(w_e|d) P_{\mathfrak{D}_e}(c|d) \quad (18)$$

$$N_{\mathfrak{D}_e}(c, \mathcal{D}_k) = \sum_{d \in \mathcal{D}_k} |d| P_{\mathfrak{D}_e}(c|d) \quad (19)$$

In addition, in Equation (13) and (14), $P_{\mathfrak{D}_e}(\mathcal{D}_k)$ can be actually viewed as the trade-off parameter modulating the degree to which EM algorithm weights the unlabelled documents translated from the language f to the language e via a bilingual lexicon. In our experiments, we assume that the constraints are satisfied, i.e. $P_{\mathfrak{D}_e}(\mathcal{D}_e) + P_{\mathfrak{D}_e}(\mathcal{D}_f) = 1$ and $P_{\mathfrak{D}_e}(d|\mathcal{D}_k) = \frac{1}{|\mathcal{D}_k|}$.

4 Experiments

4.1 Data Preparation

We chose English and Chinese as our experimental languages, since we can easily setup our experiments and they are rather different languages so that we can easily extend our algorithm to other language pairs. In addition, to evaluate the performance of our algorithm, experiments were performed over the collected data set. Standard evaluation benchmark is not available and thus we developed a test data from the Internet, containing Chinese Web pages and English Web pages. Specifically, we applied RSS reader[1] to acquire the links to the needed content and then downloaded the Web pages. Although category information of the content can be obtained by RSS reader, we still used three Chinese-English bilingual speakers to organize these Web pages into the predefined categories. As a result, the test data containing Chinese Web pages and English Web pages from various Web sites are created. The data consists of news during December 2005. Also, 5462 English Web pages are from 18 different news Web sites and 6011 Chinese Web pages are from 8 different news Web sites. Data distribution over categories is shown in Table 2. They fall into five categories: *Business*, *Education*, *Entertainment*, *Science* and *Sports*.

Some preprocessing steps are applied to Web pages. First we extract the pure texts of all Web pages, excluding anchor texts which introduce much noise. Then for Chinese corpus, all Chinese characters with BIG5 encoding first were converted into ones with GB2312 encoding, applied a Chinese segmenter tool[2] by Zhibiao Wu from LDC to our Chinese corpus and removed stop words and words with one character and less than 4 occurrences; for English corpus, we used the stop words list from SMART system (Buckley, 1985) to eliminate common words. Finally, We randomly split both the English and Chinese document collection into 75% for training and 25% for testing.

we compiled a large general-purpose English-Chinese lexicon, which contains 276,889 translation pairs, including 53,111 English entries and 38,517 Chinese entries. Actually we used a subset of the lexicon including 20,754 English entries and 13,471 Chinese entries , which occur in our corpus.

Table 2: Distribution of documents over categories

Categories	English	Chinese
Sports	1797	2375
Business	951	1212
Science	843	1157
Education	546	692
Entertainment	1325	575
Total	5462	6011

4.2 Baseline Algorithms

To investigate the effectiveness of our algorithms on cross-language text categorization, three baseline methods are used for comparison. They are denoted by ML, MT and LSI respectively.

ML (Monolingual). We conducted text categorization by training and testing the text categoriza-

[1] http://www.rssreader.com/

[2] http://projects.ldc.upenn.edu/Chinese/LDC_ch.htm

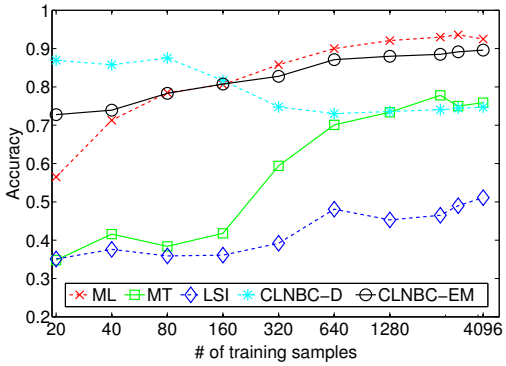


Figure 1: Comparison of the best performance of different methods with various sizes of training set and the entire test set. Training is conducted over Chinese corpus and testing is conducted over English corpus in the cross language case, while both training and testing are performed over English corpus in the monolingual case.

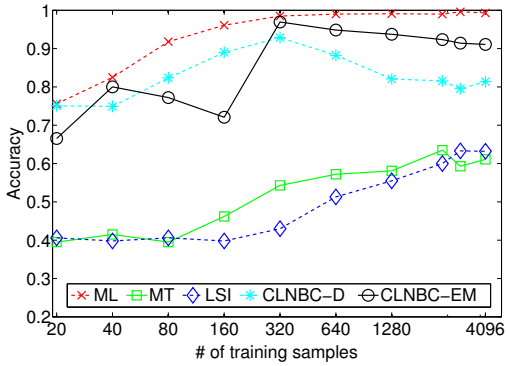


Figure 2: Comparison of the best performance of different methods with various sizes of training set and the entire test set. Training is conducted over English corpus and testing is conducted over Chinese corpus in the cross language case, while both training and testing are performed over Chinese corpus in the monolingual case.

tion system on document collection in the same language.

MT (Machine Translation). We used Systran premium 5.0 to translate training data into the language of test data, since the machine translation system is one of the best machine translation systems. Then use the translated data to learn a model for classifying the test data.

LSI (Latent Semantic Indexing). We can use the LSI or SVD technique to deduce language-independent representations through a bilingual parallel corpus. In this paper, we use SVDS command in MATLAB to acquire the eigenvectors with the first K largest eigenvalues. We take K as 400 in our experiments, where best performance is achieved.

In this paper, we use SVMs as the classifier of our baselines, since SVMs has a solid theoretic foundation based on structure risk minimization and thus high generalization ability. The commonly used one-vs-all framework is used for the multi-class case. SVMs uses the SVM^{light} software package(Joachims, 1998). In all experiments, the trade-off parameter C is set to 1.

4.3 Results

In the experiments, all results are averaged on 5 runs. Results are measured by accuracy, which is defined as the ratio of the number of labelled correctly documents to the number of all documents. When investigating how different training data have effect on performance, we randomly select the corresponding number of training samples from the training set 5 times. The results are shown in Figure 1 and Figure 2. From the two figures, we can draw the following conclusions. First, CLNBC-EM has a stable and good performance in almost all cases. Also, it can achieve the best performance among cross language methods. In addition, we notice that CLNBC-D works surprisingly better than CLNBC-EM, when there are enough test data and few training data. This may be because the quality of the probabilistic bilingual lexicon derived from CLNBC-EM method is poor, since this bilingual lexicon is trained from insufficient training data and thus may provide biased translation probabilities.

To further investigate the effect of varying the amount of test data, we randomly select the corresponding number of test samples from test set 5 times. The results are shown in Figure 3 and Figure 4, we can draw the following conclusions . First, with the increasing test data, performance of our two approaches is improved. Second, CLNBC-EM statistically significantly outperforms CLNBC-D.

From figures 1 through 4, we also notice that MT and LSI always achieve some poor results. For MT,

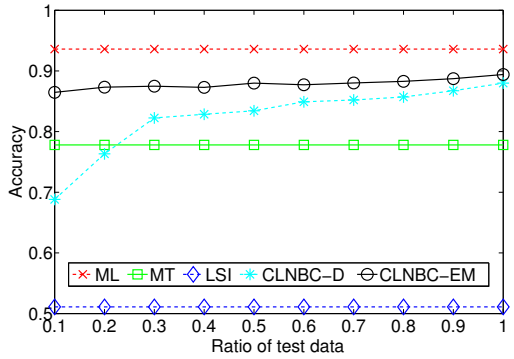


Figure 3: Comparison of the best performance of different methods with the entire training set and various sizes of test set. Training is conducted over Chinese corpus and testing is conducted over English corpus in the cross language case, while both training and testing are performed over English corpus in the monolingual case.

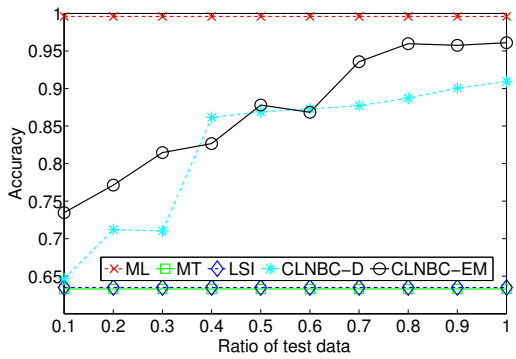


Figure 4: Comparison of the best performance of different methods with the entire training set and various sizes of test set. Training is conducted over English corpus and testing is conducted over Chinese corpus in the cross language case, while both training and testing are performed over Chinese corpus in the monolingual case.

maybe it is due to the large difference of word usage between original documents and the translated ones. For example, 骑士 (Qi Shi) has two common translations, which are *cavalier* and *knight*. In sports domain, it often means a basketball team of National Basketball Association (NBA) in U.S. and should be translated into *cavalier*. However, the translation *knight* is provided by Systran translation system we use in the experiment. In term of LSI method, one possible reason is that the parallel corpus is too limited. Another possible reason is that it is out-of-domain compared with the domain of the used document collections.

From Table 3, we can observe that our algorithm is more efficient than three baselines. The spent time are calculated on the machine, which has a 2.80GHz Dual Pentium CPU.

5 Conclusions and Future Work

In this paper, we addressed the issue of how to conduct cross language text categorization using a bilingual lexicon. To this end, we have developed a cross language naive Bayes classifier, which contains two main steps. In the first step, we deduce a probabilistic bilingual lexicon. In the second step, we adopt naive Bayes method combined with EM to conduct cross language text categorization. We have proposed two different methods, namely CLNBC-D and CLNBC-EM, for cross language text categorization. The preliminary experiments on collected data collections show the effectiveness of the proposed two methods and verify the feasibility of achieving performance near to monolingual text categorization using a bilingual lexicon alone.

As further work, we will collect larger comparable corpora to verify our algorithm. In addition, we will investigate whether the algorithm can be scaled to more fine-grained categories. Furthermore, we will investigate how the coverage of bilingual lexicon have effect on performance of our algorithm.

Table 3: Comparison of average spent time by different methods, which are used to conduct cross-language text categorization from English to Chinese.

Methods	Preparation	Computation
CLNBC-D	-	∼1 Min
CLNBC-EM	-	∼2 Min
ML	-	∼10 Min
MT	∼48 Hr[a]	∼14 Min
LSI	∼90 Min[b]	∼15 Min

[a]Machine Translation Cost
[b]SVD Decomposition Cost

Acknowledgements. The authors would like to thank three anonymous reviewers for their valuable suggestions. This work was partially supported by the National Natural Science Foundation of China under the grants NSFC 60375022 and NSFC 60473040, and the Microsoft Laboratory for Intelligent Computing and Intelligent Systems of Shanghai Jiao Tong University.

References

Nuria Bel, Cornelis H. A. Koster, and Marta Villegas. 2003. Cross-lingual text categorization. In *ECDL*, pages 126–139.

Chris Buckley. 1985. Implementation of the SMART information retrieval system. Technical report, Ithaca, NY, USA.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Transferring naive Bayes classifiers for text classification. In *Proceedings of Twenty-Second AAAI Conference on Artificial Intelligence (AAAI 2007)*, pages 540–545, July.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Pedro Domingos and Michael J. Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130.

Blaž Fortuna and John Shawe-Taylor. 2005. The use of machine translation tools for cross-lingual text mining. In *Learning With Multiple Views, Workshop at the 22nd International Conference on Machine Learning (ICML)*.

Alfio Massimiliano Gliozzo and Carlo Strapparava. 2006. Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. The Association for Computer Linguistics, July.

Thorsten Joachims. 1998. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

Cong Li and Hang Li. 2002. Word translation disambiguation using bilingual bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 343–351.

Yaoyong Li and John Shawe-Taylor. 2006. Using KCCA for Japanese-English cross-language information retrieval and document classification. *Journal of Intelligent Information Systems*, 27(2):117–133.

Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 387–394, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *Proceedings of AAAI-98, Workshop on Learning for Text Categorization*.

Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134.

J. Scott Olsson, Douglas W. Oard, and Jan Hajič. 2005. Cross-language text classification. In *Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, pages 645–646, New York, NY, August. ACM Press.

Leonardo Rigutini, Marco Maggini, and Bing Liu. 2005. An EM based training algorithm for cross-language text categorization. In *Proceedings of Web Intelligence Conference (WI-2005)*, pages 529–535, Compiègne, France, September. IEEE Computer Society.

Identify Temporal Websites Based on User Behavior Analysis

Yong Wang, Yiqun Liu,
Min Zhang, Shaoping Ma
State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China
wang-yong05@mails.thu.edu.cn

Liyun Ru
Sohu Inc. R&D center
Beijing, 100084, China
ruliyun@sohu-rd.com

Abstract

The web is growing at a rapid speed and it is almost impossible for a web crawler to download all new pages. Pages reporting breaking news should be stored into search engine index as soon as they are published, while others whose content is not time-related can be left for later crawls. We collected and analyzed into users' page-view data of 75,112,357 pages for 60 days. Using this data, we found that a large proportion of temporal pages are published by a small number of web sites providing news services, which should be crawled repeatedly with small intervals. Such temporal web sites of high freshness requirements can be identified by our algorithm based on user behavior analysis in page view data. 51.6% of all temporal pages can be picked up with a small overhead of untemporal pages. With this method, web crawlers can focus on these web sites and download pages from them with high priority.

1 Introduction

Many web users prefer accessing news reports from search engines. They type a few key words about a recent event and navigate to detailed reports about this event from the result list. Users will be frustrated if a search engine fails to perform such service and turn to other search engines to get access to news reports. In order to satisfy the users' needs, many search engines, including Google and Yahoo!, provide special channels for news retrieval and their web crawlers have to download newly appeared pages as soon as possible. However, the web is growing exponentially. The amount of new pages emerging every week is 8% of the whole web[Ntoulas et al., 2004]. It is almost impossible to download all novel pages in time.

Only a small proportion of novel pages are temporal. They report recent events and should be downloaded immediately, others which are untemporal can be downloaded later when it is convenient. So many search engines have different types of web crawlers to download the web with different policies. A common crawler checks updates of existing pages and crawls untemporal novel pages of all kinds of web sites with a relatively low frequency, usually once a month. Common crawlers are widely adopted by most search engines, but they are not suitable for news web sites which produce a great amount of pages every day. To news pages, there will be a large gap between their publication time and downloading time. Users can not get access to news pages in time. Thus another kind of crawler called instant crawler is developed. This crawler only focuses on temporal novel pages and checks updates of news web sites with much smaller intervals. Most newly-arrived content which is of high news value can be discovered by the instant crawler. Task distribution is shown in Figure 1.

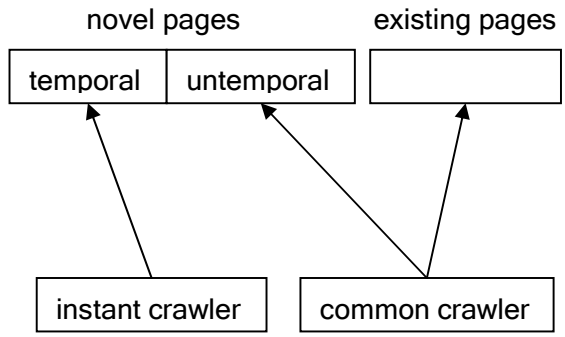


Figure 1. Job assigned to different crawlers

A relatively small set of web sites which provide news reporting services collectively generate many temporal web pages. These sites are valuable for instant crawlers and can be identified with web pages they previously generated. If a large proportion of web pages in a site are temporal, it is probable that pages published later from this the site will be temporal. Instant crawlers can focus on a list of such web sites.

Currently, the list of web sites for instant crawlers is usually generated manually, which is inevitably subjective and easily influenced by crawler administrators' preference. It includes many web sites which are actually untemporal. Also there are many mixed web sites which have both types of web pages. It is difficult for administrators to make accurate judgments about whether such sites should be included in the seed list. So instant crawlers have to spend precious and limited bandwidth to download untemporal pages while miss many temporal ones. What is more, this manually generated list is not sensitive to emerging and disappearing news sites.

In this paper, we propose a method to separate temporal pages from untemporal ones based on user behavior analysis in page-view data. Temporal web page identification is the prerequisite for temporal web site identification. A web site is temporal if most pages it publishes are temporal and most of its page-views are received from temporal pages. Then all web sites are ranked according to how temporal they are. Web sites ranked at a high position are included in the seed list for instant crawlers. Instant crawlers can focus on web sites in the list and only download pages from these web sites. Such a list covers a large proportion of temporal pages with only a small overhead of untemporal pages. The result is mined from web user behavior log, which reflects users' preference and avoids subjectivity of crawler administrators. Additionally, there are web sites associated with special events, such as Olympic Games. These web sites are temporal only when Olympic Games are being held. User behavior data can reflect the appearance and disappearance of temporal web sites.

An outline for the rest of the paper is as follows: Section 2 introduces earlier research in the evolution and discoverability of the web; Section 3 presents the user interest modal to describe web page lifetime from web users' perspective, then gives the definition of temporal web pages based on this model; Section 4 provides a method to generate a seed list for instant crawlers, and its result is also evaluated in the section; Section 5 discusses some alternatives in the experiment; Section 6 is the conclusion of this paper and suggests some possible directions in our future work.

2 Related Work

Earlier researchers performed intensive study on properties of images of the web graph[Barabasi and Albert, 1999; Broder et al., 2000; Kumar et al., 1999; Mitzenmacher, 2004]. Recently, researchers turned their attention to how the web evolves, including the rates of updates of existing pages[Brewington and Cybenko, 2000; Cho and Garcia-Molina, 2000; Fetterly et al., 2004; Pitkow and Pirolli, 1997] and the rates of new page emergence [Brewington and Cybenko, 2000]. They sent out a crawler to download web pages periodically, compared local images of the web and found characteristics of web page lifetime. Some researchers studied the frequency of web page update, predicted the lifetime of web pages and recrawl the already downloaded pages when necessary to keep the local repository fresh so that users are less bothered by stale information. They assumed that pages are modified or deleted randomly and independently with a fixed rate over time, so lifetimes of web pages are independent and identically distributed and a sequence of modifications and deletions can be modeled by a Poisson process. Other researchers focused on the discoverability of new pages[Dasgupta et al., 2007]. They tried to discover as many new pages as possible at the cost of only recrawling a few known pages.

But the web is growing explosively. It is impossible to download all the new pages. A crawler faces a frontier of the web, which is consisted of a set of discovered but not

downloaded URLs (see Figure 2). The crawler has to make a decision about which URLs should be downloaded first, which URLs should be downloaded later and which URLs are not worthy downloading at all. Thus there is some work in ordering the frontier for a crawl according to the predicted quality of the unknown pages[Cho, Garcia-Molina and Page, 1998; Eiron et al., 2004]. They predicted quality of pages which have not been downloaded yet based on the link structure of the web.

This job is similar with ours. We also make an order of the frontier, in the perspective of freshness requirements, not in the perspective of page quality. Freshness requirements differ from pages to pages. Temporal web pages whose freshness requirement timescale is minute, hour or day are assigned to the instant crawler with high priority. Other pages of lower freshness requirements can be crawled later.

This study is conducted with user behavior data instead of link structure. The link structure of the web is controlled by web site administrators. It reflects the preference of web site administrators, not that of the web users. Although in many cases, the two kinds of preference are alike, they are not identical. User behavior data reveals the real needs of web users. What is more, link structure can be easily misled by spammers. But spammers can do little to influence user behavior data contributed by mass web users.

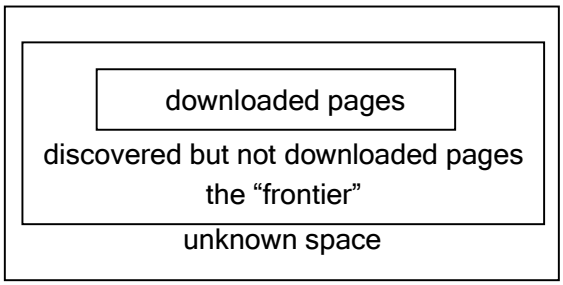


Figure 2. Web from a crawler's perspective

3 Definition of Temporal Web Page Based on Web Page Lifetime Model

3.1 Web page lifetime model

A page is born when it is published on a server, and it dies when it is deleted. But from users' view, its lifetime should not be defined by whether it is stored on a server but by whether it is accessed by users, because a web page is useful only when it can provide information for users. To web users, its life really starts at the "activation day" when the first user visits it. The page begins to be dormant on its "dormancy day" when users no longer visit it any more. After that, whether it is stored on the server does not make many differences. So the valid lifetime of a web page is the period between its activation day and its dormancy day, a subinterval of the period when it is accessible. Users' access is the only indication that the page is alive.

The state of a web page could be recorded with two values: alive and dead[Dhyani, 2002; Fetterly et al., 2003; Cho and Garcia-molina, 2003]. Its state during its valid lifetime is more complex and its liveness could be described with a continuous value: user interest. The number of page views it receives differs every day. It is more active when it is accessed by many users and it is not so active when it is accessed by fewer users. The amount of page views it receives reflects how many users are interested in it.

User interest in a web page is an amount indicating to what extent web users as a whole are interested in the page. A user visits a web page because he/she is interested in its content. The amount of page views a page receives is determined by how much user interest it can attracts. User interest in a page is a continuous variable evolving over time. User interest increases if more and more users get to know the page, and it decreases if the content is no longer fresh to users and the page becomes obsolete. User interest in a page whose content is not time related typically does not fluctuate greatly over time.

Web page lifetime could be described with user interest model, and then temporal web pages can be separated from untemporal ones according to different characteristics of their lifetime.

3.2 Definition of temporal web pages

There are two types of new pages: temporal pages and untemporal ones. Temporal pages are those reporting recent events. Users are interested in a temporal page and visit it only during a few hours or a few days after it is published. For example, a page reporting the president election result is temporal. Untemporal pages are the pages whose content is not related with recent events. There are always users visiting such pages. For example, a page introducing swimming skills is untemporal. The two kinds of new pages should be treated with different policies. The instant crawler has to download temporal pages as soon as they are discovered because users are interested in them

only in a short time span after they are born. Temporal pages are about new events and cannot be replaced by earlier pages. If the instant crawler fails to download temporal pages in time, search engine users cannot get the latest information because there are no earlier pages reporting the event which has just happened. One week after the event, even if temporal pages are downloaded, they are no longer attractive to users, just like a piece of old newspaper. In contrast, untemporal pages are not of exigencies. There is no need to download them immediately after they are published. Even if they are not downloaded in time, users can still be satisfied by other existing pages with similar content, since untemporal pages concern with problems which have already existed for a long time and have been discussed in many pages. It does not make many differences to download them early or a month later. So untemporal pages can be left to common crawlers to be downloaded later.

4 Temporal Web Sites Identification Algorithm

A seed list for an instant crawler contains temporal web sites. There are three steps to generate the seed list: search user's interest curves to describe web page lifetime; identify temporal web pages based on user interest curves; identify temporal web sites according to the proportion of temporal pages in each site.

4.1 Search User's Interest Curves

Generally speaking, few users know a newly born web page and pay attention to it. Later, more and more users get to know it, become interested in it and visit it. As time goes by, some pages become outdated and attract less user attention, while other pages never suffer from obsolescence. Users' interest in them is relatively constant. So the typical trend of user interest evolution is to increase at first then decrease in the shape of a rainbow, or to keep static. It is true that user interest in some pages experiences multi-climaxes. But it is very unlikely that those climaxes appear in our observing window of two months. Since we are studying short term web page lifetime, we do not consider user interest with multi-climaxes. The curve $y = f(x)$ that is used to describe the evolution of user interest should satisfy 4 conditions below(assuming the page is activated at time 0):

1) its field of definition is $[0, +\infty)$
2) $f(0) = 0$
3) $f(x) \geq 0$ in its field of definition
4) it has only one maximum

The probability density function (PDF) of logarithmic normal distribution is one of the functions that satisfy the conditions, so a modified edition of it, which will be addressed later, is used to describe the evolution of user interest during whole web page lifetime.

Anonymous user access log for consecutive 60 days is collected by a proxy server. Multiple requests to a single page in one day are merged as one request to avoid automatically generated large numbers of requests by spammers. Daily page view data of 75,112,357 pages from November 13th 2006 to January 11th, 2007 is recorded. Pages whose total page views during the 60 days are less than 60 (one page view each day on average) are filtered out because of lack of reliability, leaving 975,151 reliable ones. In order to retrieve user interest curves, we build a coordinate first, where the x-axis denotes time and y-axis denotes the number of page views. Given the daily page view data of a page, there are a sequence of discrete dots in the coordination (x_i, y_i), $i = 1, 2, ..., 60$, where $x_i=i$ is the *i*th day, y_i is the number of page views on the *i*th day. After that, the dots can be fitted with the formula

$$f(x) = A \times \varphi_{ln}(x) = A \times \frac{1}{\sqrt{2\pi}\sigma(x-b)} \times e^{-\frac{(\ln(x-b)-\mu)^2}{2\sigma^2}}$$

where A, b, μ, σ are parameters and $\varphi_{ln}(x)$ is the probability density function of logarithmic normal distribution. Given a page p and its page view history (x_i, y_i) $(i = 1, 2, ..., 60)$, the four parameters can be determined and the user interest curve can be defined as $y = f(x)$. One of the retrieved user interest curves is shown in Figure 3.

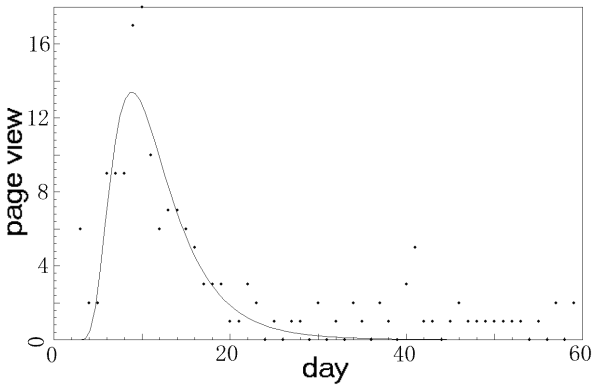


Figure 3. A User Interest Curve

4.2 Identifying Temporal Web Pages

$\varphi_{ln}(x)$ is the probability density function of a random variable. The integral of $\varphi_{ln}(x)$ in its field of definition is 1. The total user interest in a web page accumulated during its whole lifetime is

$$\int_b^{+\infty} f(x)\,dx = A \times \int_b^{+\infty} \varphi_{ln}(x)\,dx = A$$

The parameter A of a popular web page is larger than that of an unpopular one. In order to avoid discriminating popular pages and unpopular ones, parameter A for all pages is set to 1, so the area of the region enclosed by user interest curve and x-axis is 1. After this normalization, each page receives one unit user interest during their whole lifetime.

Parameter b indicates the birth time of a page. We do not care about the absolute birth time of a page, so parameter b for all pages are set to 0, which means all pages are activated at time 0.

The other two parameters σ and μ do not change, so the shape of user interest curves is reserved.

After the parameter adjusting, the user interest curve is redefined as

$$y = \varphi_{\ln}(x) = \frac{1}{\sqrt{2\pi}\sigma x} \times e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}}$$

This simpler definition of user interest curve is used in the rest of this paper.

Let

$$\Phi(x) = \int_0^x \varphi(t)\,dt$$

be the cumulative density function of logarithmic normal distribution. Given the user interest curve of page p, Φ(x) is the amount of user interest accumulated x days after its birth (see the grey area in Figure 4).

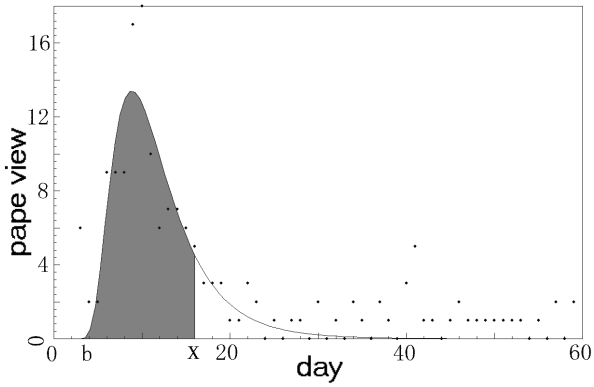


Figure 4. Accumulated user interest

A temporal web page accumulates most of its user interest during the first few days after its birth. Given a specific x, the larger Φ(x) is, the more temporal the page is, because it can accumulate more user interest during the time span. news.sohu.com is a major portal web site providing news services. Most of its pages are news reportings, which are temporal. There are 6,464 web pages from news.sohu.com and their user interest curves are retrieved. Figure 5 shows the distribution of Φ(1) of these pages. As is shown in Figure 5, on the first day of their birth, most pages have accumulated more than 80% of its total user interest of their whole lifetime. So the proportion of user interest accumulated during the beginning period of web page lifetime is a useful feature to identify temporal web pages.

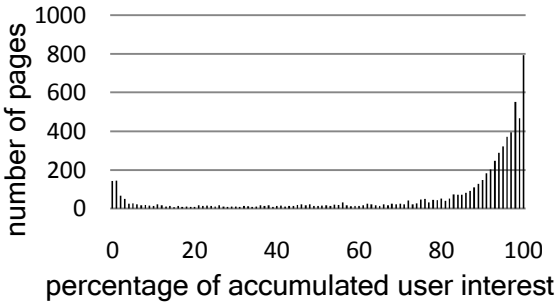


Figure 5. Distribution of accumulated user interest on the first day of birth

In order to discern temporal pages from untemporal ones, two parameters should be determined: n and q. n is the integrating range, q is the integral quantity and is also the grey area in Figure 4. Given a web page p, it is temporal if the proportion of user interest accumulated during the first n days of its lifetime is more than q (denoted in the inequality Φ(n) > q), vice versa. focus.cn is a web site about real estate. It publishes both temporal pages (such as those reporting price fluctuation information) and untemporal ones (such as those providing house decoration suggestions). We annotate 3,040 web pages in the web site of focus.cn manually, of which 2,337 are labeled temporal, 703 are labeled untemporal. After parameter adjusting, n is set to 3 and q is set to 0.7 in order to achieve the best performance that the maximized hit(the number of correct classification) is 2,829, miss(the number of temporal pages which are classified as untemporal) is 141, false alarm(the number of untemporal pages which are classified as temporal) is 70. It means that a web page will be classified as temporal if in the first three days after its birth, it can accumulate more than 70% of the total user interest it attracts during its whole lifetime.

After the classification, 135,939 web pages are labeled temporal and the other 839,212 pages are labeled untemporal.

4.3 Identifying Temporal Web Sites

A web site has many pages. There are hardly any web sites that publish temporal pages or untemporal pages exclusively. Instead, an actual web site usually contains both temporal pages and untemporal ones. For example, a web site about automobiles publishes temporal pages reporting that a new style of cars appears on the market, and it also publishes untemporal pages about how to take good care of cars. In order to classify web sites with mixed types of pages, we present definitions of temporal web sites.

From web sites administrators' view, a web site is temporal if most of its pages are temporal. So if the proportion of temporal pages of a web site in all its pages is large enough, the web site will be classified as temporal. According to this definition, the type of a web site can be controlled by its administrator. If he/she wants to make the web site temporal, he/she can publish more temporal pages. But how are these pages received by web users? Even most pages in a web site are temporal, if users pay little attention to them and are attracted mainly by untemporal ones, this web site cannot be classified as temporal. So a temporal web site should also be defined from web users' view.

From web users' view, a web site is temporal if most of its page views are received from temporal pages. Given a web site which contains both temporal pages and untemporal ones, if users are more interested in its temporal pages, the site is more likely to be classified as temporal.

Both of the two definitions above make sense. So a web site has two scores about how temporal it is based on the two definitions. The two scores are calculated in the following formulas

$$\mathrm{Score}_1(s) = \frac{\text{the number of temporal pages in s}}{\text{the number of total pages in s}}$$

$$\mathrm{Score}_2(s) = \frac{\sum_{tp \in \text{temporal pages in s}} \text{the number of page views of tp}}{\sum_{p \in \text{pages in s}} \text{the number of page views of p}}$$

where *s* is a web site. Score_1 is the proportion of temporal web pages in all pages from the web site. Score_2 is the proportion of page views received from temporal web pages in all page views to the site. Then the two scores are combined with different weights into the final score for the web site.

$$\mathrm{Score}(s) = \alpha \times \mathrm{Score}_1(s) + \beta \times \mathrm{Score}_2(s)$$

Web sites are ranked according to the final score in descending order. Search engines can pick the web sites ranked at high positions in the list as seeds for instant crawlers. They can pick as many seeds as their instant crawler is capable to monitor. In our experiment, we choose the top 100 web sites in the ranked list as temporal sites. Since there are 135,939 temporal pages and 839,212 untemporal ones in the data set, precision is defined as the proportion of temporal pages of the top 100 web sites in all pages of those sites, and recall is defined as the proportion of temporal pages of the top 100 web sites in all temporal ones in the data set. The ranked list is evaluated with the traditional IR evaluation criterion: F-Measure [Baeza-Yates and Ribeiro-Neto, 1999], which is calculated as

$$\mathrm{F-Measure} = \frac{2\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Parameter α and β are adjusted to improve F-Measure. When the ratio of α and β is 3:2, the maximized F-Measure is achieved at 0.615, where there are 70,110 temporal pages and 21,886 untemporal ones in the top 100 web sites in the ranked list.

4.4 Evaluation of the Temporal Web Site List

Human annotated results are always considered optimal in general experiment result evaluation in information retrieval. However, in our task, human annotator cannot make a perfect seed list for instant crawlers, because it is very difficult to decide whether a web site containing appropriate amount of temporal pages and untemporal ones. In contrast, the method we propose make decision not only by the proportion of temporal pages in a site, but also by how well each kind of pages are received based on the amount of page views they get. So the seed list generated from user behavior data can outperform that generated by humans.

Sohu Inc. has a manually generated list containing 100 seed web sites for its instant crawler. This list is evaluated with the method above. The 100 web sites in the list cover 59,113 temporal pages and 49,124 untemporal ones. The performance of automatic generated seed list for instant crawlers using our method is compared with that of manually generated list as the base line. The result is shown in Table 1.

Compared with the base line, the top 100 web sites in our seed list contain 18.6% more temporal pages than those in the manually generated list. The total burden of the instant crawler is also reduced by

15.0% since it downloads 16,241 less pages.

	Base line	Our method
Temporal Pages	59,113	70,110
Total Pages	108,237	91,996
Precision	54.6%	76.2%
Recall	43.5%	51.6%
F-Measure	0.484	0.615

Table 1. Evaluation of the two seed list for instant crawlers

5 Discussion

5.1 Advantages of using user interest curves

There are three advantages of using user interest curves instead of raw page-view data. First, the number of page views is determined by the amount of user interest a page receives, but they do not strictly equal. Page view data is affected by many random factors, such as whether it is weekday or weekend. These random factors are called "noise" in general. Such noise can influence the number of page views, but it is not the determinant factor. The number of page views is centered on the amount of user interest and fluctuate around it, because page view data is a combination of user interest and the noise. User interest curves are less bothered by such noise since the noise is effectively eliminated after data fitting. Second, although the observing window is two months wide, which is wide enough to cover lifetime of most temporal web pages, there are still many temporal ones whose lifetime is across the observing window boundaries. Such fragmented page view records will bring in mistakes in identifying temporal web pages. But if most part of the lifetime of a web page lies in the observing window, the data fitting process is able to estimate the absent page-view data and make up the missing part of user interest curve of its whole lifetime. So the effects brought by cross-boundary web pages can be reduced. Third, user interest curve is continuous and can be integrated to show the accumulated user interest to the page in a period of time.

5.2 Effects of using different parameter values

In our experiment, we used a single threshold n and q (see Section 5) and a page is classified as a temporal one if the user interest it accumulates during n days after its birth is greater than q. But web users receive different types of news at different speed. We notice that financial, entertainment, political and military news gets through users rapidly. These kinds of news become obsolescent to users quickly, usually only a few minutes or hours after they are published. So web pages reporting such news are ephemeral and they can draw users' attention only in a short period after their birth. In contrast, it takes much more time for users to know other kinds of news. For example, a web page reporting a volcano eruption far away from users may not be so attractive and has to spend much more time to accumulate the specific proportion of user interest. So maybe it is necessary to give different thresholds for different types of news.

In our experiment, we choose values of n and p in order to get the maximized hit (see Section 5). Some web crawlers may have abundant network bandwidth and want lower miss. Other crawlers whose network bandwidth is very limited are untolerant with false alarm. So the result of temporal page classification can be evaluated by linear combination with different weights

Performance = A × hit - B × miss - C × false alarm

Values of A, B and C can be determined according to the capacity of s crawlers.

Whether a web site is temporal is determined by the proportion of its temporal pages and the proportion of its page views received from temporal pages. The two proportions are combined with different weights α and β in order to get maximized F-Measure (see Section 6). However, to some extent, the measure of page-view proportion is misleading, because a hot event which receives a great deal of user attention is usually reported by several news agencies. It is of little value to download redundant reports from different web sites although they get many page views. Page-view data discriminates against pages reporting events which receive little attention. Most of these pages cannot be replaced by others because they are usually the only page reporting such events. Whether these pages can be correctly retrieved influence user experience greatly. Users often judge a search engine by whether the pages receiving low attention can be recalled. So the temporal page proportion should be assigned with additional weight to avoid such bias.

6 Conclusion and Future Work

The web is growing rapidly. It is impossible to download all new pages. Web crawlers have to make a decision about which pages should be downloaded with high priority. Previous

researchers made decisions according to page quality and suggested downloading pages of high quality first. They ignored the fact that temporal pages should be downloaded first. Otherwise, they will become outdated soon. It is better to download these temporal pages immediately in the perspective of freshness requirement.

Only a few web sites collectively publish a large proportion of temporal pages. In this paper, an algorithm is introduced to score each web site about how temporal it is based on page-view data which records user behavior. Web sites scored high are judged as temporal. An instant crawler can focus on temporal sites only. It can download more temporal pages and less untemporal ones in order to improve its efficiency.

Temporal web site identification can be done in finer granularity. There are several possible directions. Firstly, many web site administrators prefer distributing temporal web pages and untemporal ones in different folder. For example, pages stored under "/news/" are more likely to be temporal. Secondly, dynamic URLs (URLs that contain the character "?" and pairs of parameter and value) generated from the same web page, which are treated as different pages in the current work, are very likely to share the same timeliness. For example, if "/a.asp?p=1" is a temporal page, it is probable that "/a.asp?p=2" is temporal. In the future, we plan to study timeliness of web sites at folder level instead of site level.

References

Albert-László Barabási, Réka Albert. 1999. *Emergence of Scaling in Random Networks*, Science, Vol. 286. no. 5439, Pages 509 - 512.

Alexandros Ntoulas, Junghoo Cho, Christopher Olston. 2004. *What's new on the Web? The evolution of the Web from a search engine perspective.* Proceedings of the 13th conference on World Wide Web, Pages 1 - 12.

Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins and Janet Wiener. 2000. *Graph Structure in the Web*. Computer Networks, Volume 33, Issues 1-6, Pages 309-320.

Brian E. Brewington and George Cybenko. 2000. *How Dynamic is the Web?* Computer Networks, Volume 33, Issues 1-6, Pages 257-276.

Dennis Fetterly, Mark Manasse, Marc Najork and Janet Wiener. 2003. A Large-Scale Study of the Evolution of Web Pages. Proceedings of WWW03, Pages 669-678.

Devanshu Dhyani, Wee Keong NG, and Sourav S. Bhowmick. 2002. A Survey of Web Metrics. ACM Computing Surveys, Volume 34, Issue 4, Pages 469 – 503.

James Pitkow and Peter Pirolli. 1997. *Life, Death, and Lawfulness on the Electronic Frontier*. Proceedings of the SIGCHI conference on Human factors in computing systems, Pages 383–390, 1997.

Junghoo Cho , Hector Garcia-Molina and Lawrence Page. 1998. *Efficient Crawling Through URL Ordering*. Computer Networks, Volume 30, Number 1, Pages 161-172(12).

Junghoo Cho and Hector Garcia-Molina. 2000. *The evolution of the web and implications for an incremental crawler*. In Proc. 26th VLDB, Pages 200–209.

Junghoo Cho and Hector Garcia-Molina. 2003. *Effective Page Refresh Policies for Web Crawlers*. ACM Transactions on Database Systems, Volume 28 , Issue 4, Pages 390 – 426.

Michael Mitzenmacher. 2004. *A Brief History of Lognormal and Power Law Distributions*. Proceedings of the 39th Annual Allerton Conference on Communication, Control, and Computing.

Nadav Eiron, Kevin S. McCurley and John A. Tomlin. 2004. *Ranking the Web Frontier*. Proceedings of the 13th international conference on World Wide Web, pages 309–318.

Ravi Kumar , Prabhakar Raghavan , Sridhar Rajagopalan and Andrew Tomkins. 1999. *Trawling the Web for Emerging Cyber-communities*. Proceeding of the eighth international conference on World Wide Web, Pages 1481-1493.

Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. 1999. Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

A Comparative Study for Query Translation using Linear Combination and Confidence Measure

Youssef Kadri
Laboratoire RALI, DIRO
Université de Montréal
CP 6128, Montréal, Canada, H3C3J7
kadriyou@iro.umontreal.ca

Jian-Yun Nie
Laboratoire RALI, DIRO
Université de Montréal
CP 6128, Montréal, Canada, H3C3J7
nie@iro.umontreal.ca

Abstract

In Cross Language Information Retrieval (CLIR), query terms can be translated to the document language using Bilingual Dictionaries (BDs) or Statistical Translation Models (STMs). Combining different translation resources can also be used to improve the performance. Unfortunately, the most studies on combining multiple resources use simple methods such as linear combination. In this paper, we drew up a comparative study between linear combination and confidence measures to combine multiple translation resources for the purpose of CLIR. We show that the linear combination method is unable to combine correctly different types of resources such as BDs and STMs. While the confidence measure method is able to re-weight the translation candidate more radically than in linear combination. It reconsiders each translation candidate proposed by different resources with respect to additional features. We tested the two methods on different test CLIR collections and the results show that the confidence measure outperforms the linear combination method.

1 Introduction

Cross Language Information Retrieval (CLIR) tries to determine documents written in a language from a query written in another language. Query translation is widely considered as the key problem in this task (Oard, 1998). In previous researches, various approaches have been proposed for query translation: using a bilingual dictionary, using an off-the-shelf machine translation system or using a parallel corpus. It is also found that when multiple translation resources are used, the translation quality can be improved, comparing to using only one translation resource (Xu, 2005). Indeed, every translation tool or resource has its own limitations. For example, a bilingual dictionary can suggest common translations, but they remain ambiguous – translations for different senses of the source word are mixed up. Machine translation systems usually employ sophisticated methods to determine the best translation sentence, for example, syntactic analysis and some semantic analysis. However, it usually output only one translation for a source word, while it is usually preferred that a source query word be translated by multiple words in order to produce a desired query expansion effect. In addition, the only word choice made by a machine translation system can be wrong. Finally, parallel corpora contain useful information about word translation in particular areas. One can use such a corpus to train a statistical translation model, which can then be used to translate a query. This approach has the advantage that few manual interventions are required to produce the statistical translation model. In addition, each source word can be translated by several related target words and the latter being weighted. However, among the proposed translation words, there may be irrelevant ones.

Therefore, one can take advantage of several translation resources and tools in order to produce better query translations. The key problem is the way to combine the resources.

A common method used in previous studies is to assign a weight to each resource. Then all the translation candidates are weighted and then combined linearly (Nie, 2000). However, this kind of combination assigns a single confidence score to

all the translations from the same translation resource. In reality, a translation resource does not cover all the words with equal confidence. For some words, its translations can be accurate, while for some others, they are inappropriate. By using a linear combination, the relative order among the translation candidates is not changed. In practice, a translation with a low score can turn out to be a better translation when other information becomes available.

For example, the English word "nutritional" is translated into French by a statistical translation model trained on a set of parallel texts as follows:

{nutritive 0.32 (nutritious), alimentaire 0.21 (food)}.

We observe that the most common translation word "alimentaire" only takes the second place with lower probability than "nutritive". If these translations are combined linearly with another resource (say a BD), it is unlikely that the correct translation word "alimentaire" gain larger weight than "nutritive".

This example shows that we have to reconsider the relative weights of the translation candidates when another translation resource is available. The purpose of this reconsideration is to determine how reasonable a translation candidate is given all the information now available. In so doing, the initial ranking of translation candidates can be changed. As a matter of fact using the method of confidence measures that we propose in this paper, we are able to reorder the translation candidates as follows:

{alimentaire 0.38, nutritive 0.23, valeur 0.11 (value)}.

The weight of the correct translation "alimentaire" is considerably increased.

In this paper, we will propose to use a new method based on confidence measure to re-weight the translation candidates. In the re-weighting, the original weight according to each translation resource is only considered as one factor. The final weight is determined by combining all the available factors. In our implementation, the factors are combined in neural networks, which produce a final confidence measure for each of the translation candidates. This final weight is not a simple linear combination of the original weights, but a recalculation according to all the information available, which is not when each translation resource is estimated separately.

The advantages of this approach are twofold. On one hand, the confidence measure allows us to adjust the original weights of the translations and to select the best translation terms according to all the information. On the other hand, the confidence measures also provide us with a new weighting for the translation candidates that are comparable across different translation resources. Indeed, when we try to combine a statistical translation model with a bilingual dictionary, we had to assign a weight to a candidate from the bilingual dictionary. This weight is not directly compatible with the probability assigned in the former.

In the remaining sections of this paper, we will first describe the principle of confidence measure in section 2. In section 3, we will compare two methods to combine different translation resources: linear combination and confidence measure. Section 4 provides a description on how the parameters are tuned. Section 5 outlines the different steps for computing confidence measures. Finally, we present the results of our experiments on both English-French and English-Arabic CLIR. Our experiments will show that the method using confidence measure significantly outperforms the traditional approach using linear combination.

2 Confidence measure

Confidence measure is often used to re-rank or re-weight some outputs produced by separate means. For example, in speech recognition and understanding (Hazen et al., 2002), one tries to re-rank the result of speech recognition according to additional information using confidence measure. Gandrabur et al. (2003) used confidence measures in a translation prediction task. The goal is to re-rank the translation candidates according to additional information. Confidence measure is defined as the probability of correctness of a candidate. In the case of translation, given a candidate translation t_E for a source word t_F, the confidence measure is $P(correct \mid t_F, t_E, F)$, where F is a set of other features of the translation context (e.g. the POS-tag of the word, the previous translations words, etc.). In both applications, significant gains have been observed when using a confidence estimation layer within the translation models.

The problem of query translation is similar to general translation described in (Gandrabur et al. 2003). We are presented with several translation resources, each being built separately. Our goal now is to use all of them together. As we discussed earlier, we want to take advantage of the additional information (other translation resources as well as

additional linguistic analysis on the query) in order to re-weight each of the translation candidates.

In previous studies, neural networks have been commonly used to produce confidence measures. The inputs to the neural networks are translation candidates from different resources, their original weights and various other properties of them (e.g. POS-tag, probability in a language model, etc.). The output of the neural networks is a confidence measure assigned to a translation candidate from a translation resource. This confidence measure is used to re-rank the whole set of candidates from all the resources.

In this study, we will use the same approach to combine different translation resources and to produce confidence measures.

The neural networks need to be trained on a set of training data. Such data are available in both speech recognition and machine translation. However, in the case of CLIR, the goal of query translation is not strictly equivalent to machine translation. Indeed, in query translation, we are not limited to the correct literal translations. Not literal translation words that are strongly related to the query are also highly useful. These latter related words can produce a desired query expansion effect in IR.

Given this situation, we can no longer use a parallel corpus as our training data as in the case of machine translation. Modifications are necessary. We will describe the modified way we use to create the training data in section 4. The informative features we use will be described n section 5.2.

3 General CLIR Problem

Assume a query Q_E written in a source language E and a document D_F written in a target language F, we would like to determine a score of relevance of D_F to Q_E. However, as they are not directly comparable, a form of translation is needed. Let us describe the model that we will use to determine its score.

Various theoretical models have been developed for IR, including vector space model, Boolean model and probabilistic model. Recently, language modeling is widely used in IR, and it has been show to produce very good experimental results. In addition, language modeling also provides a solid theoretical framework for integrating more aspects in IR such as query translation. Therefore, we will use it as our basic framework in this study.

In language modeling framework, the relevance score of the document D_F to the query Q_E is determined as the negative KL-divergence between the query's language model and the document's language model (Zhai, 2001a). It is defined as follows:

$$R(Q_E, D_F) \propto \sum_{t_F} p(t_F \mid Q_E) \log p(t_F \mid D_F) \quad (1)$$

To avoid the problem of attributing zero probability to query terms not occurring in document D_F, smoothing techniques are used to estimate $p(t_F|D_F)$. One can use the Jelinek-Mercer smoothing technique which is a method of interpolating between the document and collection language models (Zhai, 2001b). The smoothed $p(t_F|D_F)$ is calculated as follows:

$$p(t_F \mid D_F) = (1-\lambda) p_{ML}(t_F \mid D_F) + \lambda p_{ML}(t_F \mid C_F) \quad (2)$$

where $p_{ML}(t_F \mid D_F) = \frac{tf(t_F, D_F)}{|D_F|}$ and $p_{ML}(t_F \mid C_F) = \frac{tf(t_F, C_F)}{|C_F|}$ are the maximum likelihood estimates of a unigram language model based on respectively the given document D_F and the collection of documents C_F. λ is a parameter that controls the influence of each model.

In CLIR, the term $p(t_F \mid Q_E)$ in equation (1) representing the query model can be estimated as follows:

$$p(t_F \mid Q_E) = \sum_{q_E} p(t_F, q_E \mid Q_E) = \sum_{q_E} p(t_F \mid q_E, Q_E) p(q_E \mid Q_E) \approx \sum_{q_E} p(t_F \mid q_E) p_{ML}(q_E \mid Q_E) \quad (3)$$

where $p_{ML}(q_E \mid Q_E)$ is the maximum likelihood estimation: $p_{ML}(q_E \mid Q_E) = \frac{tf(q_E, Q_E)}{|Q_E|}$ and $p(t_F \mid q_E)$ is the translation model. Putting (3) in (1), we obtain the general CLIR score formula:

$$R(Q_E, D_F) \propto \sum_{t_F} \sum_{q_E} p(t_F \mid q_E) p_{ML}(q_E \mid Q_E) \log p(t_F \mid D_F) \quad (4)$$

In our work, we do not change the document model $p(t_F \mid D_F)$ from monolingual IR. Our focus will be put on the estimation of the translation model $p(t_F \mid q_E)$ - the translation probability from a source query term q_E to a target word t_F, in particular, when several translation resources are available.

Let us now describe two different ways to combine different translation resources for the estimation of $p(t_F \mid q_E)$: by linear combination and by confidence measure.

4 Linear Combination

The first intuitive method to combine different translation resources is by a linear combination. This means that the final translation model is estimated as follows:

$$p(t_F \mid q_E) = z_{q_E} \sum_i \lambda_i p_i(t_F \mid q_E) \quad (5)$$

where λ_i is the parameter assigned to the translation resource *i* and z_{q_E} is a normalization factor so that $\sum_{t_F} p(t_F \mid q_E) = 1$. $p_i(t_F \mid q_E)$ is the probability of translating the source word q_E to the target word t_F by the resource *i*.

In order to determine the appropriate parameter for each translation resource, we use the EM algorithm to find values which maximize the log-likelihood LL of a set *C* of training data according to the combined model, i.e.:

$$LL(C) = \sum_{(f,e)\in C} p(f,e) \sum_{j=1}^{|f|} \log \sum_{k=1}^{n} \sum_{i=1}^{|e|} \lambda_k t_k(f_j \mid e_i) p(e_i) \quad (6)$$

Where $(f, e) \in C$ is a pair of parallel sentences; $p(f,e) = \frac{\#(f,e)}{|C|}$ is the prior probability of the pair of sentences (*f, e*) in the corpus *C*, |*f*| is the length of the target sentence *f* and |*e*| is the length of the source sentence *e*. λ_k is the coefficient related to resource *k* that we want to optimize and *n* is the number of resources. $t_k(f_j|e_i)$ is the probability of translating the source word e_i with the target word f_j with each resource. $p(e_i)$ is the prior probability of the source word e_i in the corpus *C*. Note that the validation data set *C* on which we optimize the parameters must be different from the one used to train our baseline models.

The training corpora are as follows: For English-Arabic, we use the Arabic-English parallel news corpus[1]. This corpus consists of around 83 K pairs of aligned sentences. For English-French, we use a bitext extracted from two parallel corpora: The Hansard [2] corpus and the Web corpus (Kadri, 2004). It consists of around 60 K pairs of aligned sentences.

The component models for English-Arabic CLIR are: a STM built on a set of parallel Web pages (Kadri, 2004), another STM built on the English-Arabic United Nations corpus (Fraser, 2002), Ajeeb[3] bilingual dictionary and Almisbar[4] bilingual dictionary. For English-French CLIR, we use three component models: a STM built on Hansard corpus, another STM built on parallel Web pages and the Freedict[5] bilingual dictionary.

5 Using Confidence Measures

The question considered in confidence measure is: Given a translation *candidate*, is it correct and how confident are we on its correctness?

Confidence measure aims to answer this question. Given a translation candidate t_F for a source term q_E and a set *F* of other features, confidence measure corresponds to $p_i(C=1 \mid t_F, q_E, F)$. We can use this measure as an estimate of $p(t_F \mid q_E)$, i.e.:

$$p(t_F \mid q_E) = z_{q_E} \sum_i p_i(C=1 \mid t_F, q_E, F) \quad (7)$$

where *F* is the set of features that we use. We will see several features to help determine the confidence measure of a translation candidate, for example, the translation probability, the reverse translation probability, language model features, and so on. We will describe these features in more detail in section 5.2.

In general, we can consider confidence measure as *P(C=1|X)*, given *X*— the source word, a translation and a set of features. We use a Multi Layer Perceptron (MLP) to estimate the probability of correctness *P(C=1|X)* of a translation. Neural networks have the ability to use input data of different natures and they are well-suited for classification tasks.

Our training data can be viewed as a set of pairs (*X,C*), where *X* is a vector of features relative to a translation[6] used as the input of the network, and *C* is the desired output (the correctness of the translation 0/1). The MLP implements a non-linear mapping of the input features by combining layers of linear transformation and non-linear transfer function. Formally, the MLP implements a discriminant function for an input X of the form:

[1] http://www.ldc.upenn.edu/
Arabic-English Parallel News Part 1 (LDC2004T18)

[2] LDC provides a version of this corpus: http://www.ldc.upenn.edu/.

[3] http://www.ajeeb.com/

[4] http://www.almisbar.com/

[5] http://www.freedict.com/

[6] By translation, we mean the pair of source word and its translation.

$$g(X;\theta) = o(V \times h(W \times X)) \quad (8)$$

where θ ={W,V}, W is a matrix of weights between input and hidden layers and V is a vector of weights between hidden and output layers; h is an activation function for the hidden units which non-linearly transforms the linear combination of inputs $W \times X$; o is also a non-linear activation function but for the output unit, that transforms the MLP output to the probability estimate $P(C=1|X)$. Under these conditions, our MLP was trained to minimize an objective function of error rate (Section 4.1).

In our experiments, we used a batch gradient descent optimizer. During the test stage, the confidence of a translation X is estimated with the above discriminant function $g(X; \theta)$; where θ is the set of weights optimized during the learning stage. These parameters are expected to correlate with the true probability of correctness $P(C=1|X)$.

5.1 The objective function to minimize

A natural metric for evaluating probability estimates is the negative log-likelihood (or cross entropy CE) assigned to the test corpus by the model normalized by the number of examples in the test corpus (Blatz et al., 2003). This metric evaluates the probabilities of correctness. It measures the cross entropy between the empirical distribution on the two classes (correct/incorrect) and the confidence model distribution across all the examples $X^{(i)}$ in the corpus. Cross entropy is defined as follows:

$$CE = -\frac{1}{n}\sum_{i} \log P(C^{(i)} \mid X^{(i)}) \quad (9)$$

where $C^{(i)}$ is 1 if the translation $X^{(i)}$ is correct, 0 otherwise. To remove dependence on the prior probability of correctness, Normalized Cross Entropy (NCE) is used:

$$NCE = (CE_b - CE)/CE_b \quad (10)$$

The baseline CE_b is a model that assigns fixed probabilities of correctness based on the empirical class frequencies:

$$CE_b = -(n_0/n)\log(n_0/n) - (n_1/n)\log(n_1/n) \quad (11)$$

where n_0 and n_1 are the numbers of correct and incorrect translations among n test cases.

5.2 Features

The MLP tends to capture the relationship between the correctness of the translation and the features, and its performance depends on the selection of informative features.

We selected intuitively seven classes of features hypothesized to be informative for the correctness of a translation.

Translation model index: an index representing the resource of translation that produced the translation candidate.

Translation probabilities: the probability of translating a source word with a target word. These probabilities are estimated with IBM model 1 (Brown et al., 1993) on parallel corpora. For translations from bilingual dictionaries, as no probability is provided, we carry out the following process to assign a probability to each translation pair (*e, f*) in a bilingual dictionary: We trained a statistical translation model on a parallel corpus. Then for each translation pair (*e,f*) of the bilingual dictionary, we looked up the resulting translation model and extracted the probability assigned by this translation model to the translation pair in question. Finally, the probability is normalized by the Laplace smoothing method:

$$p_{BD}(f \mid e) = \frac{p_{STM}(f \mid e) + 1}{\sum_{i=1}^{n} p_{STM}(f_i \mid e) + 1} \quad (12)$$

Where n is the number of translations proposed by the bilingual dictionary to the word e.

Translation ranking: This class of features includes two features: The rank of the translation provided by each resource and the probability difference between the translation and the highest probability translation.

Reverse translation information: This includes the probability of translation of a target word to a source word. Other features measure the rank of source word in the list of translations of the target word and if the source word holds in the best translations of the target word.

Translation "Voting": This feature aims to know whether the translation is voted by more than one resource. The more a same translation is voted the more likely it may be correct.

Source sentence-related features: One feature measures the frequency of the source word in the source sentence. Another feature measures the number of source words in the source sentence that have a translation relation with the translation in question.

Language model features: We use the unigram, the bigram and the trigram language models for source and target words on the training data.

5.3 Training for confidence measures

The corpus used for training confidence is the same as the corpus for tuning parameters for the linear combination. It is a set of aligned sentences. Source sentences are translated to the target language word by word using baseline models. We translated each source word with the most probable[7] translations for the translation models and the best five translations provided by the bilingual dictionaries. Translations are then compared to the reference sentence to build a labeled corpus: a translation of a source word is considered to be correct if it occurs in the reference sentence. The word order is ignored, but the number of occurrences is taken into account. This metric fits well our context of IR: IR models are based on "bag of words" principle and the order of words is not considered.

We test with various numbers of hidden units (from 5 to 100). We used the NCE metric to compare the performance of different architectures. The MLP with 50 hidden units gave the best performance.

To test the performance of individual features, we experimented with each class of features alone. The best features are the translation "voting", language model features and the translation probabilities. The translation "voting" is very informative because it presents the translation probability attributed by each resource to the translation in question. The translation ranking, the reverse translation information, the translation model index and the source sentence-related features provide some marginally useful information.

6 CLIR experiments

The experiments are designed to test whether the confidence measure approach is effective for query translation, and how it compares with the traditional linear combination. We will conduct two series of experiments, one for English-French CLIR and another for English-Arabic CLIR.

[7] The translations with the probability p(f|e)≥0.1

6.1 Experimental setup

English-French CLIR: We use English queries to retrieve French documents. In our experiments, we use two document collections: one from TREC[8] and another from CLEF[9] (SDA). Both collections contain newspaper articles. TREC collection contains 141 656 documents and CLEF collection 44 013 documents. We use 4 query sets: 3 from TREC (TREC6 (25 queries), TREC7 (28 queries), TREC8 (28 queries)) and one from CLEF (40 queries).

English-Arabic CLIR: For these experiments, we use English queries to retrieve Arabic documents. The test corpus is the Arabic TREC collection which contains 383 872 documents. For topics, we use two sets: TREC2001 (25 queries) and TREC2002 (50 queries).

Documents and queries are stemmed and stopwords are removed. The Porter stemming is used to stem English queries and French documents. Arabic documents are stemmed using linguistic-based stemming method (Kadri, 2006). The query terms are translated with the baseline models (Section 4). The resulting translations are then submitted to the information retrieval process. We tested with different ways to assign weights to translation candidates: translations from each resource, linear combination and confidence measures.

When using each resource separately, we attribute the IBM 1 translation probabilities to our translations. For each query term, we take only translations with the probability $p(f|e)\geq 0.1$ when using translation models and the five best translations when using bilingual dictionaries.

6.2 Linear combination (LC)

The tuned parameters assigned to each translation resource are as follows:

English-Arabic CLR:
STM-Web: 0.29, STM-UN: 0.34,
Ajeeb BD: 0.14, Almisbar BD: 0.22.

English-French CLR:
STM-Web: 0.3588, STM-Hansard: 0.6408,
Freedict BD: 0.0003.

These weights produced the best log-likelihood of the training data.

[8] http://trec.nist.gov/

[9] http://www.clef-campaign.org/

For CLIR, the above combinations are used to combine translation candidates from different resources. The tables below show the CLIR effectiveness (mean average precision - MAP) of individual models and the linear combination.

Translation Model	TREC 2001	TREC 2002	Merged TREC 2001/2002
Monolingual IR	(0.33)	(0.28)	(0.31)
STM-Web	0.14 (42%)	0.04 (17%)	0.07 (25%)
STM-UN	0.11 (33%)	0.09 (34%)	0.10 (33%)
Ajeeb BD	0.27 (81%)	0.19 (70%)	0.22 (70%)
Almisbar BD	0.17 (51%)	0.16 (58%)	0.16 (54%)
Linear Comb.	0.24 (72%)	0.20 (71%)	0.21 (67%)

Table1. English-Arabic CLIR performance (MAP) with individual models and linear combination

Trans. Model	TREC6	TREC7	TREC8	CLEF
Monolingual IR	0.39	0.34	0.44	0.40
STM-Web	0.22 (56%)	0.17 (50%)	0.22 (50%)	0.29 (72%)
STM-Hansard	0.25 (64%)	0.24 (70%)	0.33 (75%)	0.30 (75%)
Freedict BD	0.17 (43%)	0.11 (32%)	0.13 (29%)	0.14 (35%)
Linear Comb.	0.26 (66%)	0.26 (76%)	0.36 (81%)	0.30 (75%)

Table2. English-French CLIR performance (MAP) with individual models and linear combination

We observe that the performance is quite different from one model to another. The low score recorded by the STMs for English-Arabic CLIR compared to the score of STMs for English-French CLIR is possibly due to the small data set on which the English-Arabic STMs are trained. A set of 2816 English-Arabic pairs of documents is not enough to build a reasonable STM. For English-Arabic CLIR, BDs present better performance than STMs because they cover almost all query terms and they provide multiple good translations to each query term. When combining all the resources, the performance is supposed to be better because we would like to take advantage of each of the models. However, we see that the combined model performs even worse than one of the models - Ajeeb BD for English-Arabic CLIR. This shows that the linear combination is not necessarily a good way to combine different translation resources.

An example of English queries is shown in Table 3: "What measures are being taken to develop tourism in Cairo?". The Arabic translation provided by TREC to the word "measures" is: "إجراءات". We see clearly that translations with different resources are different. Some resources propose inappropriate translations such as "مكيال" or "ميزان". Even if two resources suggest the same translations, the weights are different. For this query, the linear combination produces better query translation terms than every resource taken alone: The most probable translations are selected from the combined list. However, this method is unable to attribute an appropriate weight to the best translation "إجراءات"; it is selected but ranked at third position with a weak weight.

Trans. model	Translation(s) of word "measures"
Ajeeb BD	0.05 (measure) تدبير, 0.05 (caliber) عيار, 0.05 (measurement) مقياس, 0.05 (measurement) مقياس, 0.05 (standard) معيار, 0.05 (standard) مكيال, 0.05 (balance) ميزان
Almisbar BD	0.05 (procedures) إجراءات, 0.03 مقياس, 0.03 قدر (measurement), 0.03 (amount) مقدار
STM-UN	0.69 (measures) تدابير
STM-Web	0.09 إجراءات
Linear Comb,	0.61 تدابير, 0.037 مقياس, 0.029 إجراءات, قياس 0.020

Table3. Translation examples

6.3 CLIR with Confidence Measures (CM)

In these experiments, we use confidence measures as weights for translations. According to these confidence measures, we select the translations with the best confidences for each query term. The following tables show the results:

Collection	TREC 2001	TREC 2002	TREC01-02
MAP of LC	0.2426	0.2032	0.2163
MAP of CM	0.2775(14.35%)	0.2052 (1%)	0.2290 (5.87 %)

Table4. Comparison of English-Arabic CLIR between linear combination and confidence measures

Collection	TREC6	TREC7	TREC8	CLEF
MAP of LC	0.2692	0.2630	0.3605	0.3071
MAP of CM	0.2988 (10.99%)	0.2699 (2.62%)	0.3761 (4.32%)	0.3230 (5.17 %)

Table5. Comparison of English-French CLIR between linear combination and confidence measures

In terms of MAP, we see clearly that the results using confidence measures are better than those obtained with the linear combination. The two-tailed t-test shows that the improvement brought by confidence measure over linear combination is statistically significant at the level $P<0.05$. This improvement in CLIR performance is attributed to the ability of confidence measure to re-weight each translation candidate. The final sets of translations (and their probabilities) are more reasonable than in linear combination. The tables below show some examples where we get a large improvement in average precision when using confidence measures to combine resources. The first example is the TREC 2001 query "What measures are being taken to develop tourism in Cairo?". The translation of the query term "measures" to Arabic using the two

methods is presented in table 6. The second example is the TREC6 query "Acupuncture". Table 7 presents the translation of this query term is to French using the two techniques:

Trans.Model	Translation(s) of term "measures"
Linear Comb.	تدابير 0.61, مقياس 0.037, إجراءات 0.029, قياس 0.020
Conf. meas.	إجراءات 0.51, قدر 0.10, قياس 0.06

Table6. Translation examples to Arabic

Trans.model	Translation(s) of term "Acupuncture"
Linear Comb.	Acupuncture 0.13 (acupuncture), sevrage 0.13 (severing), hypnose 0.13 (hypnosis)
Conf. meas.	Acupuncture 0.21, sevrage 0.17, hypnose 0.14

Table7. Translation examples to French

In the example of table 6, confidence measure has been able to redeem the best translation "إجراءات" and rescore it with a stronger weight than the other incorrect or inappropriate ones. The same effect is observed in the example of table 7. Confidence measure has been able to increase the correct translation "acupuncture" to a higher level than the other incorrect ones. These examples show the potential advantage of confidence measure over linear combination: The confidence measure does not blindly trust all the translations from different resources. It tests their validity on new validation data. Thus, the translation candidates are rescored and filtered according to a more reliable weight.

7 Conclusion

Multiple translation resources are believed to contribute in improving the quality of query translation. However, in most previous studies, only linear combination has been used. In this study, we propose a new method based on confidence measure to combine different translation resources. The confidence measure estimates the probability of correctness of a translation, given a set of features available. The measure is used to weight the translation candidates in a unified manner. It is also expected that the new measure is more reasonable than the original measures because of the use of additional features. Our experiments on both English-Arabic and English-French CLIR have shown that confidence measure is a better way to combine translation resources than linear combination. This shows that confidence measure is a promising approach to combine non homogenous resources and can be further improved on several aspects. For example, we can optimize this technique by identifying other informative features. Other techniques for computing confidence estimates can also be used in order to improve the performance of CLIR.

References

J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis and N. Ueffing. 2003. *Confidence estimation for machine translation*. Technical Report, CLSP/JHU 2003 Summer Workshop, Baltimore MD.

P. F. Brown, S. A. Pietra, V. J. Pietra and R. L. Mercer. 1993. *The mathematics of statistical machine translation: Parameter estimation*. Computational Linguistics, 19(2):263–311.

A. Fraser, J. Xu and R. Weischedel. 2002. *TREC 2002 Cross-lingual Retrieval at BBN*. TREC11 conference.

S. Gandrabur and G. Foster. 2003. *Confidence Estimation for Text Prediction*. Proceedings of the CoNLL 2003 Conference, Edmonton.

T. J. Hazen, T. Burianek, J. Polifroni and S. Seneff. 2002. *Recognition confidence scoring for use in speech understanding systems*. Computer Speech and Language, 16:49-67.

Y. Kadri and J. Y. Nie. 2004. *Query translation for English-Arabic cross language information retrieval*. Proceedings of the TALN conference.

Y. Kadri and J. Y. Nie. 2006. *Effective stemming for Arabic information retrieval*. The challenge of Arabic for NLP/MT Conference. The British Computer Society. London, UK.

J. Y. Nie, M. Simard and G Foster. 2000. *Multilingual information retrieval based on parallel texts from the Web*. In LNCS 2069, C. Peters editor, CLEF2000:188-201, Lisbon.

D. W. Oard and A. Diekema. 1998. *Cross-Language Information Retrieval*. In M. Williams (ed.), Annual review of Information science, 1998:223-256.

J. Xu and R. Weischedel. 2005. *Empirical studies on the impact of lexical resources on CLIR performance*. Information processing & management, 41(3):475-487.

C. Zhai and J. Lafferty. 2001a. *Model-based feedback in the language modeling approach to information retrieval*. CIKM 2001 Conference.

C. Zhai and J. Lafferty. 2001b. *A study of smoothing methods for language models applied to ad hoc information retrieval*. Proceedings of the ACM-SIGIR.

TSUBAKI: An Open Search Engine Infrastructure for Developing New Information Access Methodology

Keiji Shinzato[†], **Tomohide Shibata**[†], **Daisuke Kawahara**[‡], **Chikara Hashimoto**[‡‡] and **Sadao Kurohashi**[†]

[†]Graduate School of Informatics, Kyoto University
[‡]National Institute of Information and Communications Technology
[‡‡]Department of Informatics, Yamagata University

{shinzato, shibata, kuro}@nlp.kuee.kyoto-u.ac.jp
dk@nict.go.jp ch@yz.yamagata-u.ac.jp

Abstract

As the amount of information created by human beings is explosively grown in the last decade, it is getting extremely harder to obtain necessary information by conventional information access methods. Hence, creation of drastically new technology is needed. For developing such new technology, search engine infrastructures are required. Although the existing search engine APIs can be regarded as such infrastructures, these APIs have several restrictions such as a limit on the number of API calls. To help the development of new technology, we are running an open search engine infrastructure, TSUBAKI, on a high-performance computing environment. In this paper, we describe TSUBAKI infrastructure.

1 Introduction

As the amount of information created by human beings is explosively grown in the last decade (University of California, 2003), it is getting extremely harder to obtain necessary information by conventional information access methods, i.e., Web search engines. This is obvious from the fact that knowledge workers now spend about 30% of their day on only searching for information (The Delphi Group White Paper, 2001). Hence, creation of drastically new technology is needed by integrating several disciplines such as natural language processing (NLP), information retrieval (IR) and others.

Conventional search engines such as Google and Yahoo! are insufficient to search necessary information from the current Web. The problems of the conventional search engines are summarized as follows:

Cannot accept queries by natural language sentences: Search engine users have to represent their needs by a list of words. This means that search engine users cannot obtain necessary information if they fail to represent their needs into a proper word list. This is a serious problem for users who do not utilize a search engine frequently.

Cannot provide organized search results: A search result is a simple list consisting of URLs, titles and snippets of web pages. This type of result presentation is obviously insufficient considering explosive growth and diversity of web pages.

Cannot handle synonymous expressions: Existing search engines ignore a synonymous expression problem. Especially, since Japanese uses three kinds of alphabets, Hiragana, Katakana and Kanji, this problem is more serious. For instance, although both Japanese words "こども" and "子供" mean *child*, the search engines provide quite different search results for each word.

We believe that new IR systems that overcome the above problems give us more flexible and comfortable information access and that development of such systems is an important and interesting research topic.

To develop such IR systems, a search engine infrastructure that plays *a low-level layer role* (i.e., retrieving web pages according to a user's query from a huge web page collection) is required. The Application Programming Interfaces (APIs) provided by

commercial search engines can be regarded as such search engine infrastructures. The APIs, however, have the following problems:

1. The number of API calls a day and the number of web pages included in a search result are limited.
2. The API users cannot know how the acquired web pages are ranked because the ranking measure of web pages has not been made public.
3. It is difficult to reproduce previously-obtained search results via the APIs because search engine's indices are updated frequently.

These problems are an obstacle to develop new IR systems using existing search engine APIs.

The research project "Cyber Infrastructure for the Information-explosion Era[1]" gives researchers several kinds of shared platforms and sophisticated tools, such as an open search engine infrastructure, considerable computational environment and a grid shell software (Kaneda et al., 2002), for creation of drastically new IR technology. In this paper, we describe an open search engine infrastructure **TSUBAKI**, which is one of the shared platforms developed in the Cyber Infrastructure for the Information-explosion Era project. The overview of TSUBAKI is depicted in Figure 1. TSUBAKI is built on a high-performance computing environment consisting of 128 CPU cores and 100 tera-byte storages, and it can provide users with search results retrieved from approximately 100 million Japanese web pages.

The mission of TSUBAKI is to help the development of new information access methodology which solves the problems of conventional information access methods. This is achieved by the following TSUBAKI's characteristics:

API without any restriction: TSUBAKI provides its API without any restrictions such as the limited number of API calls a day and the number of results returned from an API per query, which are the typical restrictions of the existing search engine APIs. Consequently, TSUBAKI API users can develop systems that handle a large number of web pages. This feature is important for dealing with the Web that has the long tail aspect.

[1]http://i-explosion.ex.nii.ac.jp/i-explosion/ctr.php/m/IndexEng/a/Index/

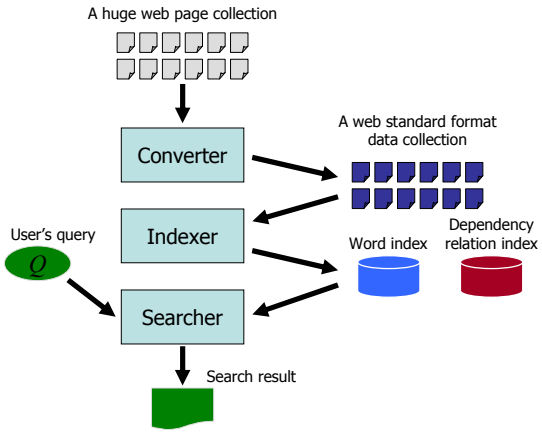


Figure 1: An overview of TSUBAKI.

Transparent and reproducible search results: TSUBAKI makes public not only its ranking measure but also its source codes, and also provides reproducible search results by fixing a crawled web page collection. Because of this, TSUBAKI keeps its architecture transparency, and systems using the API can always obtain previously-produced search results.

Web standard format for sharing pre-processed web pages: TSUBAKI converts a crawled web page into a web standard format data. The web standard format is a data format used in TSUBAKI for sharing pre-processed web pages. Section 2 presents the web standard format in detail.

Indices generated by deep NLP: TSUBAKI indexes all crawled web pages by not only words but also dependency relations for retrieving web pages according to the meaning of their contents. The index data in TSUBAKI are described in Section 3.

This paper is organized as follows. Section 2 describes web standard format, and Section 3 shows TSUBAKI's index data and its search algorithm. Section 4 presents TSUBAKI API and gives examples of how to use the API. Section 5 shows related work.

2 Sharing of Pre-processed Web Pages on a Large Scale

Web page processing on a large scale is a difficult task because the task generally requires a

high-performance computing environment (Kawahara and Kurohashi, 2006) and not everybody can use such environment. Sharing of large scale pre-processed web pages is necessary for eliminating the gap yielded by large data processing capabilities.

TSUBAKI makes it possible to share pre-processed large scale web pages through the API. TSUBAKI API provides not only cached original web pages (i.e., 100 million pages) but also pre-processed web pages. As pre-processed data of web pages, the results of commonly performed processing for web pages, including sentence boundary detection, morphological analysis and parsing, are provided. This allows API users to begin their own processing immediately without extracting sentences from web pages and analyzing them by themselves.

In the remainder of this section, we describe a web standard format used in TSUBAKI for sharing pre-processed web pages and construction of a large scale web standard format data collection.

2.1 Web Standard Format

The web standard format is a simple XML-styled data format in which meta-information and text-information of a web page can be annotated. The meta-information consists of a title, in-links and out-links of a web page and the text-information consists of sentences extracted from the web page and their analyzed results by existing NLP tools.

An example of a web standard format data is shown in Figure 2. Extracted sentences are enclosed by <RawString> tags, and the analyzed results of the sentences are enclosed by <Annotation> tags. Sentences in a web page and their analyzed results can be obtained by looking at these tags in the standard format data corresponding to the page.

2.2 Construction of Web Standard Format Data Collection

We have crawled 218 million web pages over three months, May - July in 2007, by using the Shim-Crawler,[2] and then converted these pages into web standard format data with results of a Japanese parser, KNP (Kurohashi and Nagao, 1994), through our conversion tools. Note that this web page collection consists of pages written not only in Japanese but also in other languages.

```
<?xml version="1.0" encoding="UTF-8"?>
<StandardFormat
Url="http://www.kantei.go.jp/jp/koizumiprofile/1_sinnen.html" OriginalEncoding="Shift_JIS" Time="2006-08-14 19:48:51"><Text Type="default">
<S Id="1" Length="70" Offset="525">
  <RawString>小泉総理の好きな格言のひとつに「無信不立（信無くば立たず）」があります。</RawString>
  <Annotation Scheme="KNP">
    <![CDATA[* 1D <文頭><サ変><人名><助詞><連体修飾><体言><係:ノ格><区切:0-4><RID:1056>
小泉 こいずみ 小泉 名詞 6 人名 5 * 0 * 0 NIL <文頭><漢字><かな漢字><名詞相当語><自立><タグ単位始><文節始><固有キー>
 ...
ます ます ます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 基本形 2 NIL <表現文末><かな漢字><ひらがな><活用語><付属><非独立無意味接尾辞>
。 。 。 特殊 1 句点 1 * 0 * 0 NIL <文末><記号><付属>
    EOS]]>
  </Annotation>
</S>
...
</Text>
</StandardFormat>
```

Figure 2: An example of web standard format data with results of the Japanese parser KNP.

The web pages in the collection are converted into the standard format data according to the following four steps:

Step 1: Extract Japanese web pages from a given page collection.

Step 2: Detect Japanese sentence boundaries in the extracted web pages.

Step 3: Analyze the Japanese sentences by the NLP tools.

Step 4: Generate standard format data from the extracted sentences and their analyzed results.

We followed the procedure proposed in Kawahara and Kurohashi (2006) for Steps 1 and 2.

The web pages were processed by a grid computing environment that consists of 640 CPU cores and 640 GB main memory in total. It took two weeks to finish the conversion. As a result, 100 million web standard format data were obtained. In other words, the remaining 118 million web pages were regarded as non-Japanese pages by our tools.

The comparison between original web pages and the standard format data corresponding to these pages in terms of file size are shown in Table 1. We

[2] http://www.logos.t.u-tokyo.ac.jp/crawler/

Table 1: File size comparison between original web pages and standard format data (The number of web pages is 100 millions, and both the page sets are compressed by gzip.)

Document set	File size [TB]
Original web pages	0.6
Standard format styled data	3.1

can see that the file size of the web standard format data is over five times bigger than that of the original web pages.

3 Search Engine TSUBAKI

In this section, we describe the indices and search algorithm used in TSUBAKI.

3.1 Indices in TSUBAKI

TSUBAKI has indexed 100 million Japanese web pages described in Section 2.2. Inverted index data were created by both words and dependency relations. Note that the index data are constructed from parsing results in the standard format data.

3.1.1 Word Index

Handling of synonymous expressions is a crucial problem in IR. Especially, since Japanese uses three kinds of alphabets, Hiragana, Katakana and Kanji, spelling variation is a big obstacle. For example, the word "*child*" can be represented by at least three spellings "こども", "子ども" and "子供" in Japanese. Although these spellings mean *child*, existing search engines handle them in totally different manner. Handling of spelling variations is important for improving search engine performance.

To handle spelling variations properly, TSUBAKI exploits results of JUMAN (Kurohashi et al., 1994), a Japanese morphological analyzer. JUMAN segments a sentence into words, and gives representative forms of the words simultaneously. For example, JUMAN gives us "子供" as a representative form of the words "こども", "子ども" and "子供." TSUBAKI indexes web pages by word representative forms. This allows us to retrieve web pages that include different spellings of the queries.

TSUBAKI also indexes word positions for providing search methods such as an *exact phrase search*. A word position reflects the number of words appearing before the word in a web page. For example, if a page contains N words, the word appearing in the beginning of the page and the last word are assigned 0 and $N - 1$ as their positions respectively.

3.1.2 Dependency Relation Index

The understanding of web page contents is crucial for obtaining necessary information from the Web. The word frequency and link structure have been used as clues for conventional web page retrieval. These clues, however, are not sufficient to understand web page's contents. We believe that other clues such as parsing results of web page contents are needed for the understanding.

Let us consider the following two sentences:

S1: Japan exports automobiles to Germany.

S2: Germany exports automobiles to Japan.

Although the above sentences have different meanings, they consist of the same words. This means that a word index alone can never distinguish the semantic difference between these sentences.

On the other hand, syntactic parsers can produce different dependency relations for each sentence. Thus, the difference between these sentences can be grasped by looking at their dependency relations. We expect that dependency relations work as efficient clues for understanding web page contents.

As a first step toward web page retrieval considering the meaning of web page contents, TSUBAKI indexes web pages by not only words but also dependency relations. An index of the dependency relation between *A* and *B* is represented by the notation $A \rightarrow B$, which means *A modifies B*. For instance, the dependency relation indices *Japan* $\rightarrow$ *export*, *automobile* $\rightarrow$ *export*, *to* $\rightarrow$ *export*, and *Germany* $\rightarrow$ *to* are generated from the sentence **S1**.

3.1.3 Construction of Index data

We have constructed word and dependency relation indices from a web standard format data collection described in Section 2.2. The file size of the constructed indices are shown in Table 2. We can see that the file size of the word index is larger than that of dependency relation. This is because that the word index includes all position for all word index expression.

Table 2: File sizes of the word and dependency relation indices constructed from 100 million web pages.

Index type	File size [TB]
Word	1.17
Dependency relation	0.89

Table 3: Comparison with index data of TSUBAKI and the Apache Lucene in terms of index data size (The number of web pages is a million.)

Search engine	File size [GB]
TSUBAKI (words)	12.0
TSUBAKI (dependency relations)	9.1
Apache Lucene	4.7

Moreover, we have compared index data constructed by TSUBAKI and the Apache Lucene,[3] an open source information retrieval library, in terms of the file size. We first selected a million web pages from among 100 million pages, and then indexed them by using the indexer of TSUBAKI and that of the Lucene.[4] While TSUBAKI's indexer indexed web pages by the both words and dependency relations, the Lucene's indexer indexed pages by only words. The comparison result is listed in Table 3. We can see that the word index data constructed by TSUBAKI indexer is larger than that of the Lucene. But, the file size of the TSUBAKI's index data can be made smaller because the TSUBAKI indexer does not optimize the constructed index data.

3.2 Search Algorithm

TSUBAKI is run on a load balance server, four master servers and 27 search servers. Word and dependency relation indices generated from 100 million web pages are divided into 100 pieces respectively, and each piece is allocated to the search servers. In short, each search server has the word and dependency relation indices generated from at most four million pages.

The procedure for retrieving web pages is shown in Figure 3. Each search server calculates relevance scores between a user's query and each document that matches the query. We used the sum of OKAPI BM25 (Robertson et al., 1992) scores over index expressions in the query as the relevance score. The relevance score $score_{rel}$ is defined as:

$$score_{rel}(Q, d) = \sum_{q \in Q} BM25(q, d)$$

$$BM25(q, d) = w \times \frac{(k_1 + 1)fq}{K + fq} \times \frac{(k_3 + 1)qfq}{k_3 + qfq}$$

$$w = \log \frac{N - n + 0.5}{n + 0.5}, K = k_1((1 - b) + b\frac{l}{l_{ave}})$$

where q is an index expression extracted from the query Q, fq is the frequency of the expression q in a document d, qfq is the frequency of q in Q, and N is the total number of crawled web pages. TSUBAKI used 1.0×10^8 as N. n is the document frequency of q in 100 million pages, l is the document length of d (we used the number of words in the document d), and l_{ave} is the average document length over all the pages. In addition to them, the parameters of OKAPI BM25, k_1,k_3 and b were set to 2, 0 and 0.75, respectively.

Consider the expression "*global warming's effect*" as a user's query Q. The extracted index expressions from Q are shown in Figure 4. Each search server calculates a BM25 score for each index expression (i.e., *effect*, *global*, ... , *global* $\rightarrow$ *warm*), and sums up the calculated scores.

Note that BM25 scores of dependency relations are larger than those of single words because the

Step 1: The load balance server forwards user's query Q to the most unoccupied master server.

Step 2: The master server extracts the set of index expressions q from the given query Q, and transmits the set of q and search conditions such as a logical operator (i.e., AND/OR) between words in Q to 27 search servers.

Step 3: The search server retrieves web pages according to the set of q and search conditions by using word and dependency relation indices.

Step 4: The search server calculates a relevance score for each retrieved document, and then returns the documents with their scores to the master server.

Step 5: The master server sorts the returned documents according to their calculated scores.

Step 6: The top M documents are presented to the user as a search result.

Figure 3: The search procedure of TSUBAKI. (Steps 3 and 4 are performed in parallel.)

[3] http://lucene.apache.org/java/docs/index.html

[4] We used the Lucene 2.0 for Japanese which is available from https://sen.dev.java.net/servlets/ProjectDocumentList?folderID=755&ex pandFolder=755&folderID=0

Word index: *effect*, *global*, *warm*,
Dependency relation index: *global* → *warm*, *warm* → *effect*

Figure 4: The index expressions extracted from the query "*global warming's effect.*"

document frequencies of dependency relations are relatively smaller than those of single words. Consequently, TSUBAKI naturally gives high score values to web pages that include the same dependency relations as the one included in the given query.

4 TSUBAKI API

As mentioned before, TSUBAKI provides the API without any restriction. The API can be queried by "REST (Fielding, 2000)-Like" operators in the same way of Yahoo! API. TSUBAKI API users can obtain search results through HTTP requests with URL-encoded parameters. Examples of the available request parameters are listed in Table 4. The sample request using the parameters is below:

Case 1: Get the search result ranked at top 20 with snippets for the search query "京都 (*Kyoto*)".
http://tsubaki.ixnlp.nii.ac.jp/api.cgi?query=%E4%BA%AC%E9%83%BD&starts=1&results=20

TSUBAKI API returns an XML document in Figure 5 for the above request. The result includes a given query, a hitcount, the IDs of web pages that match the given query, the calculated scores and others. The page IDs in the result enable API users to obtain cached web pages and web standard format data. An example request for obtaining the web standard format data with document ID 01234567 is below.

Case 2: Get web standard format data with the document ID 01234567.
http://tsubaki.ixnlp.nii.ac.jp/api.cgi?id=01234567&format=xml

The hitcounts of words are frequently exploited in NLP tasks. For example, Turney (Turney, 2001) proposed a method that calculates semantic similarities between two words according to their hitcounts obtained from an existing search engine. Although TSUBAKI API users can obtain a query's hitcount from a search result shown in Figure 5, TSUBAKI API provides an access method for directly obtaining query's hitcount. The API users can obtain only a hitcount according to the following HTTP request.

Table 4: The request parameters of TSUBAKI API.

Parameter	Value	Description
query	*string*	The query to search for (UTF-8 encoded). The query parameter is required for obtaining search results.
start	*integer*: default 1	The starting result position to return.
results	*integer*: default 20	The number of results to return.
logical_operator	AND/OR: default AND	The logical operation to search for.
dpnd	0/1: default 1	Specifies whether to use dependency relations as clues for document retrieving. Set to 1 to use dependency relations.
only_hitcounts	0/1: default 0	Set to 1 to obtain a query's hitcount only.
snippets	0/1: default 0	Set to 1 to obtain snippets.
id	*string*	The document ID to obtain a cached web page or standard format data corresponding to the ID. This parameter is required for obtaining web pages or standard format data.
format	html/xml	The document type to return. This parameter is required if the parameter id is set.

Case 3: Get the hitcount of the query "京都 (*Kyoto*)"
http://tsubaki.ixnlp.nii.ac.jp/api.cgi?query=%E4%BA%AC%E9%83%BD&only_hitcounts=1

In this case, the response of the API is a plain-text data indicating the query's hitcount.

5 Related Work

As mentioned before, existing search engine APIs such as Google API are insufficient for infrastructures to help the development of new IR methodology, since they have some restrictions such as a limited number of API calls a day. The differences between TSUBAKI API and existing search engine APIs are summarized in Table 5. Other than access restrictions, the serious problem of these APIs is that they cannot always reproduce previously-provided

```
<ResultSet time="2007-10-15 14:27:01" query="京都" totalResultsAvailable="4721570" totalResultsReturned="20"
          firstResultPosition="1" logicalOperator="AND" forceDpnd="0" dpnd="1" filterSimpages="1">
   <Result Rank="1" Id="017307147" Score="8.87700">
      <Title>ＪＴＢ　ｅ－Ｈｏｔｅｌの京都府のホテル・旅館一覧</Title>
      <Url>http://www.docch.net/blog/jtb-e/kyouto.shtml</Url>
      <Snippet/>
      <Cache>
         <Url>http://tsubaki.ixnlp.nii.ac.jp/index.cgi?URL=INDEX_DIR/h017/h01730/017307147.html&KEYS=%E4%BA%
              AC%E9%83%BD</Url>
         <Size>2900</Size>
      </Cache>
   </Result>
 ...
</ResultSet>
```

Figure 5: An example of a search result returned from TSUBAKI API.

search results because their indices are updated frequently. Because of this, it is difficult to precisely compare between systems using search results obtained on different days. Moreover, private search algorithms are also the problem since API users cannot know what goes on in searching web pages. Therefore, it is difficult to precisely assess the contribution of the user's proposed method as long as the method uses the existing APIs.

Open source projects with respect to search engines such as the Apache Lucene and the Rast[5] can be also regarded as related work. Although these projects develop an open search engine module, they do not operate web search engines. This is different from our study. The comparison between TSUBAKI and open source projects with respect to indexing and ranking measure are listed in Table 6.

The Search Wikia project[6] has the similar goal to one of our goals. The goal of this project is to create an open search engine enabling us to know how the system and the algorithm operate. However, the algorithm of the search engine in this project is not made public at this time.

The Web Laboratory project (Arms et al., 2006) also has the similar goal to ours. This project aims at developing an infrastructure to access the snapshots of the Web taken by the Internet Archive.[7] Currently the pilot version of the infrastructure is released. The released infrastructure, however, allows users to access only the web pages in the Amazon.com Web site. Therefore, TSUBAKI is different from the infrastructure of the Web Laboratory project in terms of the scale of a used web page collection.

[5] http://projects.netlab.jp/rast/

[6] http://search.wikia.com/wiki/Search_Wikia

[7] http://www.archive.org/index.php

Table 5: The differences between TSUBAKI API and existing search engine APIs.

Features	Google	Yahoo!	TSUBAKI
# of API calls a day	1,000	50,000	unlimited
# of URLs in a search result	1,000	1,000	unlimited
Providing cached pages	Yes	Yes	Yes
Providing processed pages	No	No	Yes
Updating indices	Yes	Yes	No

Table 6: Comparison with indexing and ranking measure.

Search Engine	Indexing	Ranking Measure
TSUBAKI	word, dependency relation	OKAPI BM25
Apache Lucene	character bi-gram, word	TF·IDF
RAST	character bi-gram, word	TF·IDF

6 Conclusion

We have described TSUBAKI, an open search engine infrastructure for developing new information access methodology. Its major characteristics are:

- the API without any restriction,
- transparent and reproducible search results,
- Web standard format for sharing pre-processed web pages and
- indices generated by deep NLP.

TSUBAKI provides not only web pages retrieved from 100 million Japanese pages according to a user's query but also pre-processed large scale web

pages produced by using a high-performance computing environment.

On the TSUBAKI infrastructure, we are developing a new information access method that organizes retrieved web pages in a search result into clusters of pages that have relevance to each other. We believe that this method gives us more flexible information access than existing search methods.

Furthermore, we are building on the TSUBAKI infrastructure a common evaluation environment to evolve IR methodology. Such an environment is necessary to easily evaluate novel IR methodology, such as a new ranking measure, on a huge-scale web collection.

Our future work is to handle synonymous expressions such as "*car*" and "*automobile*." Handling synonymous expressions is important for improving the performance of search engines. The evaluation of TSUBAKI's performance is necessary, which is also our future work.

References

Wiiliam Y. Arms, Selcuk Aya, Pavel Dmitriev, Blazej J. Kot, Ruth Mitchell, and Lucia Walle. 2006. Building a research library for the history of the web. In *Proceedings of the Joint Conference on Digital Libraries, June 2006*, pages 95–102.

Roy Thomas Fielding. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. thesis, University of California, Irvine.

Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. 2002. Virtual private grid: A command shell for utilizing hundreds of machines efficiently. In *In 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002)*.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 1344–1347.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, (4):507–534.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of japanese morphological analyzer juman. In *The International Workshop on Sharable Natural Language*, pages 22 – 28.

Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. 1992. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30.

The Delphi Group White Paper. 2001. Connecting to your knowledge nuggets. http://www.delphiweb.com/knowledgebase/documents/upload/pdf/1802.pdf.

Peter Turney. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pages 491–502.

University of California. 2003. How much information? 2003. http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/.

A Study on Effectiveness of Syntactic Relationship in Dependence Retrieval Model

Fan Ding[1,2]
1: Graduate University,
Chinese Academy of Sciences
Beijing, 100080, China
dingfan@ict.ac.cn

Bin Wang[2]
2: Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, 100080, China
wangbin@ict.ac.cn

Abstract

To relax the Term Independence Assumption, Term Dependency is introduced and it has improved retrieval precision dramatically. There are two kinds of term dependencies, one is defined by term proximity, and the other is defined by linguistic dependencies. In this paper, we take a comparative study to re-examine these two kinds of term dependencies in dependence language model framework. Syntactic relationships, derived from a dependency parser, Minipar, are used as linguistic term dependencies. Our study shows: 1) Linguistic dependencies get a better result than term proximity. 2) Dependence retrieval model achieves more improvement in sentence-based verbose queries than keyword-based short queries.

1 Introduction

For the sake of computational simplicity, Term Independence Assumption (TIA) is widely used in most retrieval models. It states that terms are statistically independent from each other. Though unreasonable, TIA did not cause very bad performance. However, relaxing the assumption by adding term dependencies into the retrieval model is still a basic IR problem. Relaxing TIA is not easy because improperly relaxing may introduce much noisy information which will hurt the final performance. Defining the term dependency is the first step in dependence retrieval model. Two research directions are taken to define the term dependency. The first is to treat term dependencies as term proximity, for example, the Bi-gram Model (F. Song and W. B. Croft, 1999) and Markov Random Field Model (D. Metzler and W. B. Croft, 2005) in language model. The second direction is to derive term dependencies by using some linguistic structures, such as POS block (Lioma C. and Ounis I., 2007) or Noun/Verb Phrase (Mitra et al., 1997), Maximum Spanning Tree (C. J. van Rijsbergen, 1979) and Linkage Model (Gao et al., 2004) etc.

Though linguistic information is intensively used in QA (Question Answering) and IE (Information Extraction) task, it is seldom used in document retrieval (T. Brants, 2004). In document retrieval, how effective linguistic dependencies would be compared with term proximity still needs to be explored thoroughly.

In this paper, we use syntactic relationships derived by a popular dependency parser, Minipar (D. Lin, 1998), as linguistic dependencies. Minipar is a broad-coverage parser for the English language. It represents the grammar as a network of nodes and links, where the nodes represent grammatical categories and the links represent types of dependency. We extract the dependencies between content words as term dependencies.

To systematically compare term proximity with syntactic dependencies, we study the dependence retrieval models in language model framework and present a smooth-based dependence language model (SDLM). It can incorporate these two kinds of term dependencies. The experiments in TREC collections show that SDLM with syntactic relationships achieves better result than with the term proximity.

The rest of this paper is organized as follows. Section 2 reviews some previous relevant work,

Section 3 presents the definition of term dependency using syntactic relationships derived by Minipar. Section 4 presents in detail the smooth-based dependence language model. A series of experiments on TREC collections are presented in Section 5. Some conclusions are summarized in Section 6.

2 Related Work

Generally speaking, when using term dependencies in language modeling framework, two problems should be considered: The first is to define and identify term dependencies; the second is to integrate term dependencies into a weighting schema. Accordingly, this section briefly reviews some recent relevant work, which is summarized into two parts: the definition of term dependencies and weight of term dependencies.

2.1 Definition of Term Dependencies

In definition of term dependencies, there are two main methods: shallow parsing by some linguistic tools and term proximity with co-occurrence information. Both queries and documents are represented as a set of terms and term dependencies among terms. Table 1 summarizes some recent related work according to the method they use to identify term dependencies in queries and documents.

Methods	Document Parsing	Document Proximity
Query Parsing	I: DM,LDM, etc.	II: CULM, RP, etc.
Query Proximity	III: NIL	IV: BG ,WPLM, MRF, etc.

Table 1. Methods in identifying dependencies

In the part I of table 1, DM is Dependence Language Model (Gao et al., 2004). It introduces a dependency structure, called linkage model. The linkage structure assumes that term dependencies in a sentence form an acyclic, planar graph, where two related terms are linked. LDM (Gao et al., 2005) represents the related terms as linguistic concepts, which can be semantic chunks (e.g. named entities like person name, location name, etc.) and syntactic chunks (e.g. noun phrases, verb phrases, etc.).

In the part II of table 1, CULM (M. Srikanth and R. Srihari, 2003) is a concept unigram language model. The parser tree of a user query is used to identify the concepts in the query. Term sequence in a concept is treated as bi-grams in the document model. RP (Recognized Phrase, S. Liu et al., 2004) uses some linguistic tools and statistical tools to recognize four types of phrase in the query, including proper names, dictionary phrase, simple phrase and complex phrase. A phrase is in a document if all its content words appear in the document within a certain window size. The four kinds of phrase correspond to variant window size.

In the part IV of table 1, BG (bi-gram language model) is the simplest model which assumes term dependencies exist only between adjacent words both in queries and documents. WPLM (word pairs in language model, Alvarez et al., 2004) relax the co-occurrence window size in documents to 5 and relax the order constraint in bi-gram model. MRF (Markov Random Field) classify the term dependencies in queries into sequential dependence and full dependence, which respectively corresponds to ordered and unordered co-occurrence within a predefine-sized window in documents.

From above discussion we can see that when the query is sentence-based, parsing method is preferred to proximity method. When the query is keyword-based, proximity method is preferred to parsing method. Thorsten (T. Brants, 2004) note: the longer the queries, the bigger the benefit of NLP. This conclusion also holds for the definition of query term dependencies.

2.2 Weight of Term Dependencies

In dependence retrieval model, the final relevance score of a query and a document consists of both the independence score and dependence score, such as Bahadur Lazarsfeld expansion (R. M. Losee, 1994) in classical probabilistic IR models. However, Spark Jones et al. point out that without a theoretically motivated integration model, documents containing dependencies (e.g. phrases) may be over-scored if they are weighted in the same way as single words (Jones et al., 1998). Smoothing strategy in language modeling framework provide such an elegant solution to incorporate term dependencies.

In the simplest bi-gram model, the probability of bi-gram (q_{i-1},q_i) in document D is smoothed by its unigram:

$$P_{smoothed}(q_i \mid q_{i-1}, D) = \lambda \times P(q_i \mid D) + (1-\lambda) \times P(q_i \mid q_{i-1}, D)$$
$$where, P(q_i \mid q_{i-1}, D) \equiv \frac{P(q_{i-1} q_i \mid D)}{P(q_{i-1} \mid D)} \quad (1)$$

Further, the probability of bi-gram (q_{i-1},q_i) in document $P(q_i|q_{i-1},D)$ can be smoothed by its probability in collection $P(q_i|q_{i-1},C)$. If $P(q_i|q_{i-1},D)$ is smoothed as Equation (1), the relevance score of query $Q=\{q_1q_2\ldots q_m\}$ and document D is:

$$\log P(Q \mid D) = \log P(q_1 \mid D) + \sum_{i=2..m} \log P_{smoothed}(q_i \mid q_{i-1}, D)$$
$$= \log P(q_1 \mid D) + \sum_{i=2..m} \log(\lambda \times P(q_i \mid D) + (1-\lambda) \times P(q_i \mid q_{i-1}, D))$$
$$= \sum_{i=1...m} \log P(q_i \mid D) + \sum_{i=2...m} \log(\lambda + (1-\lambda) \times \frac{P(q_i \mid q_{i-1}, D)}{P(q_i \mid D)}) \quad (2)$$
$$\propto \sum_{i=1...m} \log P(q_i \mid D) + \sum_{i=2...m} \log(1 + \frac{1-\lambda}{\lambda} \times \frac{P(q_{i-1} q_i \mid D)}{P(q_i \mid D) \times P(q_{i-1} \mid D)})$$
$$= \sum_{i=1...m} \log P(q_i \mid D) + \sum_{i=2...m} MI_{smoothed}(q_{i-1}, q_i \mid D)$$
$$usually\, MI(q_{i-1}, q_i \mid D) \equiv \log \frac{P(q_{i-1} q_i \mid D)}{P(q_i \mid D) \times P(q_{i-1} \mid D)}$$

In Equation (2), the first score term is independence unigram score and the second score term is smoothed dependence score. Usually λ is set to 0.9, i.e., the dependence score is given a less weight compared with the independence score.

DM (Gao et al., 2004), which can be regarded as the generalization of the bi-gram model, gives the relevance score of a document as:

$$\log P(Q \mid D) = \sum_{i=1...m} \log P(q_i \mid D) + \log P(L \mid D) + \sum_{(i,j) \in L} MI(q_i, q_j \mid L, D) \quad (3)$$

In Equation (3),L is the set of term dependencies in query Q. The score function consists of three parts: a unigram score, a smoothing factor $\log P(L|D)$, and a dependence score $MI(q_i,q_j|L,D)$.

MRF (D. Metzler and W. B. Croft, 2005) combines the score of full independence, sequential dependence and full dependence in an interpolated way with the weight (0.8, 0.1, 0.1).

Though these above models are derived from different theories, smoothing is an important part when incorporating term dependencies.

3 Syntactic Parsing of Queries and Documents

Term dependencies defined as term proximity may contain many "noisy" dependencies. It's our belief that parsing technique can filter out some of these noises and syntactic relationship is a clue to define term dependencies. We use a popular dependency parser, Minipar, to extract the syntactic dependency between words. In this section we will discuss the extraction of syntactic dependencies and the indexing schemes of term dependencies.

3.1 Extraction of Syntactic Dependencies

A dependency relationship is an asymmetric binary relationship between a word called head (or governor, parent), and another word called modifier (or dependent, daughter). Dependency grammars represent sentence structures as a set of dependency relationships. For example, Figure 1 takes the description field of TREC topic 651 as an example and shows part of the parsing result of Minipar.

TREC Topic 651: "How is the ethnic make-up of the U.S. population changing?"

	Node1	Cat1:Rel:Cat2	Node2
…			
3	makeup	N:det:Det	the
4	makeup	N:mod:A	ethnic
5	makeup	N:lex-mod:U	make
6	makeup	N:lex-mod:U	-
8	makeup	N:mod:Prep	of
11	of	Prep:pcomp-n:N	population
9	population	N:det:Det	the
10	population	N:nn:N	U.S.
…			

Figure 1. Parsing Result of Minipar

In Figure 1, Cat is the lexical category of word, and Rel is a label assigned to the syntactic dependencies, such as subject (sub), object (obj), adjunct (mod:A), prepositional attachment (Prep:pcomp-n), etc. Since function words have no meaning, the dependency relationships including function words, such as N:det:Det, are ignored. Only the dependency relationships between content words are extracted. However, prepositional attachment is an exception. A prepositional noun phrase contains two parts: (N:mod:Prep) and (Prep:pcomp-n:N). We combine these two parts and get a relationship between nouns.

Mostly, the nodes in the parsing result are single words. When the nodes are proper names, dictionary phrases, or compound words connected by hyphen, there are more than one word in the node. For example, the 5th and 6th relationship in Figure 1 describes a compound word "make up". We divide these nodes into bi-grams, which assume dependencies exist between adjacent words inside the

nodes. If the compound-word node has a relationship with other nodes, each word in the compound-word node is assumed to have a relationship with the other nodes. Finally, the term dependencies are represented as word pairs. The direction of syntactic dependencies is ignored.

3.2 Indexing of Term Dependencies

Parsing is a time-consuming process. And the documents parsing should be an off-line process. The parsing results, recognized as term dependencies, should be organized efficiently to support the computation of relevance score at the retrieval step. As a supplement of regular documents↔words inverted index, the indexing of term dependencies is organized as documents→dependencies lists. For example, Document A has n unique words; each of these n words has relationships with at least one other word. Then the term dependencies inside these n words can be represented as a half-angle matrix as Figure 2 shows.

$$\begin{array}{c} \\ \mathrm{tid}_1 \\ \mathrm{tid}_2 \\ \cdots \\ \mathrm{tid}_{n-1} \\ \mathrm{tid}_n \end{array} \begin{array}{c} \mathrm{tid}_1\ \mathrm{tid}_2\ \ldots\ \mathrm{tid}_{n-1}\ \mathrm{tid}_n \\ \left\{ \begin{array}{ccccc} 0 & 1 & ... & 0 & 2 \\ * & 0 & .. & 5 & 4 \\ * & * & ... & 3 & 0 \\ * & * & ... & 0 & 1 \\ * & * & * & * & 0 \end{array} \right\} \end{array}$$

Figure 2. Half-angle matrix of term dependencies

The (i,j)-th element of the matrix is the number of times that tid_i and tid_j have a dependency in document A. The matrix has the size of (n-1)*n/2 and it is stored as list of size (n-1)*n/2. Each document corresponds to such a matrix. When accessing the term dependencies index, the global word id in the regular index is firstly converted to the internal id according to the word's appearance order in the document. The internal id is the index of the half-angle matrix. Using the internal id pair, we can get its position in the matrix.

4 Smooth-based Dependence Model

From the discussion in section 2.2, we can see that smoothing is very important not only in unigram language model, but also in dependence language model. Taking the smoothed unigram model (C. Zhai and J. Lafferty, 2001) as the example, the retrieval status value (RSV) has the form:

$$RSV_{UG}(Q,D) = \sum_{w \in Q \cap D} c(w,Q) \log \frac{p_{DML}(w|D)}{\alpha_D p(w|C)} + |Q| \log \alpha_D \quad (4)$$

In Equation (4), c(w,Q) is the frequency of *w* in Q. The equation has three parts: $P_{DML}(w|D)$, α_D and P(w|C). $P_{DML}(w|D)$ is the discounted maximum likelihood estimation of unigram P(w|D), α_D is the smoothing coefficient of document D, and P(w|C) is collection language model. If we use a smoothing strategy as the smoothed MI in Equation (2), and replace term *w* with term pair (w_i,w_j), we can get the smoothed dependence model as:

$$RSV_{DEP}(Q,D) = \sum_{(w_i,w_j) \in L \cap D} c(w_i,w_j,Q) \log(1 + \lambda_0 \times \frac{p_{smooth}(w_i,w_j|D)}{p_{smooth}(w_i,w_j|C)}) \quad (5)$$

In Equation (5), λ_0 is the smoothing coefficient. $P_{smooth}(w_i,w_j|D)$ and $P_{smooth}(w_i,w_j|C)$ is the smoothed weight of term pair (w_i,w_j) in document D and collection C.

4.1 Smoothing $P(w_i,w_j|D)$

We use two parts to estimate the $P_{smooth}(w_i,w_j|D)$: one is the weight of the term pair with relationships in D, $P(w_i,w_j|R,D)$, the other is the weight of the term co-occurrence in D, $P_{co}(w_i,w_j|D)$. These two parts are defined as below:

$$\begin{aligned} &P(w_i,w_j|R,D) = C_D(w_i,w_j,R)/|D| \\ &P_{CO}(w_i,w_j|D) = \sqrt{P(w_i|D) \times P(w_j|D)} \\ &P(w_i|D) = C_D(w_i)/|D| \end{aligned} \quad (6)$$

|D| is the document length, $C_D(w_i,w_j,R)$ denotes the count of the dependency (w_i,w_j) in the document D, and $C_D(w_i)$ is the frequency of word w_i in D. $P_{smooth}(w_i,w_j|D)$ is defined as a combination of the two parts:

$$\begin{aligned} &P_{smooth}(w_i,w_j|D) = \lambda_1 \times P(w_i,w_j|R,D) \\ &+ (1-\lambda_1) \times P_{co}(w_i,w_j|D) \end{aligned} \quad (7)$$

4.2 Smoothing $P(w_i,w_j|C)$

To directly estimate the probability of term pair (w_i,w_j) in the collection is not easy. We use document frequency of term pair (w_i,w_j) as its approximation. Same as $P_{smooth}(w_i,w_j|D)$, $P_{smooth}(w_i,w_j|C)$ consists of two parts: one is the document frequency of term pair (w_i,w_j), $DF(w_i,w_j)$, the other is the averaged document frequency of w_i and w_j. Then, $P_{smooth}(w_i,w_j|C)$ is defined as:

$$\begin{aligned} &P_{smooth}(w_i,w_j|C) = \lambda_2 \times DF(w_i,w_j)/|C|_D \\ &+ (1-\lambda_2) \times \sqrt{DF(w_i) \times DF(w_j)}/|C|_D \end{aligned} \quad (8)$$

In Equation (8), $|C|_D$ is the count of Document in Collection C.

Finally, if substituting Equation (7) and (8) into Equation (5), there are three parameters $(\lambda_0,\lambda_1,\lambda_2)$ in $RSV_{DEP}(Q,D)$. The final retrieval status value of the smooth-based dependence model, RSV_{SDLM}, is the sum of RSV_{DEP} and RSV_{UG}:

$$RSV_{SDLM}(Q,D) = RSV_{DEP}(Q,D) + RSV_{UG}(Q,D) \quad (9)$$

5 Experiments and Results

To answer the question whether the syntactic dependencies is more effective than term proximity, we systematically compared their performance on two kinds of queries. One is verbose queries (the description field of TREC topics), the other is short queries (the title field of TREC topics). Since the verbose queries are sentence-level, they are parsed by Minipar to get the syntactic dependencies. In short queries, term proximity is used to define the dependencies, which assume every two words in the queries have a dependency.

Our smooth-based dependence language model (SDLM) is used as dependence retrieval model in the experiments. If defining $C_D(w_i,w_j,R)$ in Equation (6) to different meanings, we can get a dependence model with syntactic dependencie, SDLM_Syn, or a dependence model with term proximity, SDLM_Prox. In SDLM_Syn, $C_D(w_i,w_j,R)$ is the count of syntactic dependencies between w_i and w_j in D. In SDLM_Prox, $C_D(w_i,w_j,R)$ is the number of times the terms w_i and w_j appear within a window N terms.

We use Dirichlet-Prior smoothed KL-Divergence model as the unigram model in Equation (9). The Dirichlet-Prior smoothing parameter is set to 2000. This unigram model, UG, is also the baseline in the experiments. The main evaluation metric in this study is the non-interpolated average precision (AvgPr.)

We evaluated the smooth-based dependence language model in two document collections and four query collections. Some statistics of the collections are shown in Table 2.

Three retrieval models are evaluated in the TREC collections: UG, SDLM_Syn and SDLM_Prox. Besides the parameters $(\lambda_0,\lambda_1,\lambda_2)$, SDLM_Prox has one more parameter than SDLM_Syn. It is the window size N of $C_D(w_i,w_j,R)$. In the experiments, we tried the window size N of 5, 10, 20 and 40 to find the optimal setting. We find the optimal N is 10. This size is close to sentence length and it is used in the following experiments.

Coll.	Queries	Documents	Size (MB)	# Doc.
AP	51-200	Associated Press (1988,1989) in Disk2	489	164,597
TREC7-8	351-450	Disk 4&5 (no CR)	3,120	528,155
Robust04 Hard	35 hard queries in 351-450			
Robust04 New	651-700 ex.672			

Table 2. TREC collections

Parameters $(\lambda_0,\lambda_1,\lambda_2)$ were trained on three query sets: 51-200, 351-450 and 651-700. Each query set was divided into two halves, and we applied two-fold cross validation to get the final result. We trained $(\lambda_0,\lambda_1,\lambda_2)$ by directly maximizing MAP (mean average precision). Since the parameter range was limited, we used a linear search method at step 0.1 to find the optimal setting of $(\lambda_0,\lambda_1,\lambda_2)$.

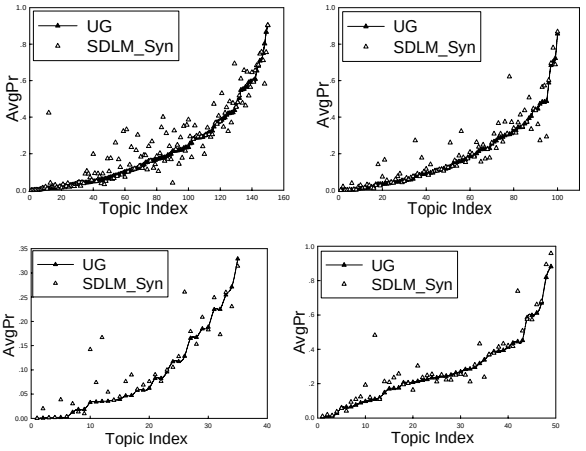


Figure 3 UG vs. SDLM_Syn in verbose queries: Top Left (51-200), Top Right (351-450), Bottom Left (hard topics in 351-450), and Bottom Right (651-700)

The results on verbose queries and short queries are listed in Table 3 and Table 4 respectively. The settings of $(\lambda_0,\lambda_1,\lambda_2)$ used in the experiments are also listed. A star mark after the change percent value indicates a statistical significant difference at the 0.05 level(one-sided Wilcoxon test). In verbose queries, we can see that SDLM has distinct

collections	UG	SDLM_Prox			SDLM_Syn		
	AvgPr.	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)
AP	0.2159	0.2360	9.31*	(1.8,0.6,0.9)	0.2393	10.84*	(1.9,0.7,0.9)
TREC7-8	0.1893	0.2049	8.24*	(1.2,0.1,0.2)	0.2061	8.87*	(0.4,0.1,0.9)
Robust04_hard	0.0909	0.1049	15.40*	(1.2,0.1,0.2)	0.1064	17.05*	(0.4,0.1,0.9)
Robust04_new	0.2754	0.3022	9.73*	(0.7,0.1,0.3)	0.3023	9.77*	(0.7,0.1,0.3)

Table 3. Comparison results on verbose queries

collections	UG	SDLM_Prox			SDLM_Syn		
	AvgPr.	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)	AvgPr.	%ch over UG	(λ_0,λ_1,λ_2)
AP	0.2643	0.2644	0	(1.3,0.6,0.1)	0.2647	0.15	(1.1,0.5,0.2)
TREC7-8	0.2069	0.2076	0.34	(1.2,0.3,0.2)	0.2070	0	(1,0.1,0.2)
Robust04_hard	0.1037	0.1044	0.68	(1.2,0.3,0.2)	0.1045	0.77	(1,0.1,0.2)
Robust04_new	0.2771	0.2888	4.22*	(1.3,0.3,0.4)	0.2869	3.54*	(1.3,0.1,0.4)

Table 4. Comparison results on short queries

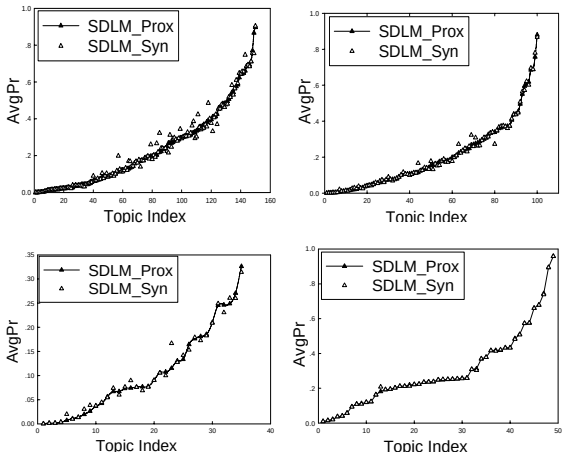


Figure 4. SDLM_Prox vs. SDLM_Syn in verbose queries: Top Left (51-200), Top Right (351-450), Bottom Left (hard topics in 351-450), Bottom Right (651-700)

improvement over UG and SDLM_Syn has robust improvement over SDLM_Prox. In short queries, SDLM has slight improvement over UG and SDLM_Syn is comparative with SDLM_Prox.

To study the effectiveness of syntactic dependencies in detail, Figure 3 and 4 compare SDLM_Syn and UG, SDLM_Syn and SDLM_Prox topic by topic in verbose queries.

As shown in Figure 3 and Figure 4, SDLM_Syn achieves substantial improvements over UG in the majority of queries. While SDLM_Syn is comparative with SDLM_Prox in most of the queries, SDLM_Syn still get some noticeable improvements over SDLM_Prox.

From Table 3 and 4, we can see while the parameters (λ_0,λ_1,λ_2) change a lot in two different document collections, there is little change in the same document collection. This shows the robustness of our smooth-based dependence language model.

6 Conclusion

In this paper we have systematically studied the effectiveness of syntactic dependencies compared with term proximity in dependence retrieval model. To compare the effectiveness of syntactic dependencies and term proximity, we develop a smooth-based dependence language model that can incorporate different term dependencies.

Experiments on four TREC collections indicate the effectiveness of syntactic dependencies: In verbose queries, the improvement of syntactic dependencies over term proximity is noticeable; In short queries, the improvement is not noticeable. For keywords-based short queries with average length of 2-3 words, the term dependencies in the queries are very few. So the improvement of dependence retrieval model over independence unigram model is very limited. Meanwhile, the difference between syntactic dependencies and term proximity is not noticeable. For dependence retrieval model, we can get the same conclusion as Thorsten Brants: the longer the queries are, the bigger the benefit of NLP is.

References

C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

Carmen Alvarez, Philippe Langlais, Jian-Yun Nie, *Word Pairs in Language Modeling for Information Retrieval*, In Proceedings of RIAO 2004, Pages 686-705, 2004

Chengxiang Zhai and John Lafferty. *A study of smoothing methods for language models applied to ad hoc information retrieval*. In Proceedings of SIGIR'01, pages 334–342, 2001

Dekang Lin, *Dependency-based Evaluation of MINIPAR*, Proceedings of Workshop on the Evaluation of Parsing Systems, Granada, Spain, May, 1998.

Donald Metzler and W. Bruce Croft, *A Markov random field model for term dependencies*, In Proceedings of SIGIR'05, Pages 472-479, 2005

Fei Song and W. Bruce Croft. *A general language model for information retrieval.* In Proceedings of SIGIR'99, pages 279-280, 1999.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu and Guihong Cao, *Dependence Language Model for Information Retrieval*, In Proceedings of SIGIR'04, Pages:170-177, 2004

Jianfeng Gao, Haoliang Qi, Xinsong Xia and Jian-Yun Nie. *Linear Discriminant Model for Information Retrieval*. In Proceedings of SIGIR'05, Pages:290-297, 2005

K. Sparck Jones, S. Walker, and S. E. Robertson, *A probabilistic model of information retrieval: development and status*. Technical Report TR-446, Cambridge University Computer Laboratory. 1998

Lioma C. and Ounis I., *A Syntactically-Based Query Reformulation Technique for Information Retrieval*, *Information Processing and Management (IPM)*, Elsevier Science, 2007

M. Mitra, C. Buckley, A. Singhal, and C. Cardie. *An Analysis of Statistical and Syntactic Phrases.* In Proceedings of RIAO-97, 5th International Conference "Recherche d'Information Assistee par Ordinateur", pages 200-214, Montreal, CA, 1997.

Munirathnam Srikanth and Rohini Srihari, *Exploiting Syntactic Structure of Queries in a Language Modeling Approach to IR*, In Proceedings of CIKM'03, Pages: 476-483, 2003

Shuang Liu, Fang Liu, Clement Yu and Weiyi Meng, *An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases*, In Proceedings of SIGIR'04, Pages: 266-272, 2004

Robert M. Losee. *Term dependence: Truncating the Bahadur Lazarsfeld expansion*. Information Processing and Management, 30(2):293–303, 1994.

Thorsten Brants. *Natural Language Processing in Information Retrieval* . In Proceedings of 20th International Conference on Computational Linguistics, Antwerp, Belgium, 2004:1-13.

Automatic Estimation of Word Significance oriented for Speech-based Information Retrieval

Takashi Shichiri
Graduate School of Science and Tech.
Ryukoku University
Seta, Otsu 520-2194, Japan
shichiri@nlp.i.ryukoku.ac.jp

Hiroaki Nanjo
Faculty of Science and Tech.
Ryukoku University
Seta, Otsu 520-2194, Japan
nanjo@nlp.i.ryukoku.ac.jp

Takehiko Yoshimi
Faculty of Science and Tech.
Ryukoku University
Seta, Otsu 520-2194, Japan
yoshimi@nlp.i.ryukoku.ac.jp

Abstract

Automatic estimation of word significance oriented for speech-based Information Retrieval (IR) is addressed. Since the significance of words differs in IR, automatic speech recognition (ASR) performance has been evaluated based on weighted word error rate (WWER), which gives a weight on errors from the viewpoint of IR, instead of word error rate (WER), which treats all words uniformly. A decoding strategy that minimizes WWER based on a Minimum Bayes-Risk framework has been shown, and the reduction of errors on both ASR and IR has been reported. In this paper, we propose an automatic estimation method for word significance (weights) based on its influence on IR. Specifically, weights are estimated so that evaluation measures of ASR and IR are equivalent. We apply the proposed method to a speech-based information retrieval system, which is a typical IR system, and show that the method works well.

1 Introduction

Based on the progress of spoken language processing, the main target of speech processing has shifted from speech recognition to speech understanding. Since speech-based information retrieval (IR) must extract user intention from speech queries, it is thus a typical speech understanding task. IR typically searches for appropriate documents such as newspaper articles or Web pages using statistical matching for a given query. To define the similarity between a query and documents, the word vector space model or "bag-of-words" model is widely adopted, and such statistics as the TF-IDF measure are introduced to consider the significance of words in the matching. Therefore, when using automatic speech recognition (ASR) as a front-end of such IR systems, the significance of the words should be considered in ASR; words that greatly affect IR performance must be detected with higher priority.

Based on such a background, ASR evaluation should be done from the viewpoint of the quality of mis-recognized words instead of quantity. From this point of view, word error rate (WER), which is the most widely used evaluation measure of ASR accuracy, is not an appropriate evaluation measure when we want to use ASR systems for IR because all words are treated identically in WER. Instead of WER, weighted WER (WWER), which considers the significance of words from a viewpoint of IR, has been proposed as an evaluation measure for ASR. Nanjo et.al showed that the ASR based on the Minimum Bayes-Risk framework could reduce WWER and the WWER reduction was effective for key-sentence indexing and IR (H.Nanjo et al., 2005).

To exploit ASR which minimizes WWER for IR, we should appropriately define weights of words. Ideal weights would give a WWER equivalent to IR performance degradation when a corresponding ASR result is used as a query for the IR system. After obtaining such weights, we can predict IR degradation by simply evaluating ASR accuracy, and thus, minimum WWER decoding (ASR) will be the most effective for IR.

For well-defined IRs such as relational database retrieval (E.Levin et al., 2000), significant words (=keywords) are obvious. On the contrary, determining significant words for more general IR task (T.Misu et al., 2004) (C.Hori et al., 2003) is not easy. Moreover, even if significant words are given, the weight of each word is not clear. To properly and easily integrate the ASR system into an IR system, the weights of words should be determined automatically. Conventionally, they are determined by an experienced system designer. Actually, in conventional studies of minimum WWER decoding for key-sentence indexing (H.Nanjo and T.Kawahara, 2005) and IR (H.Nanjo et al., 2005), weights were defined based on TF-IDF values used in back-end indexing or IR systems. These values reflect word significance for IR, but are used without having been proven suitable for IR-oriented ASR. In this paper, we propose an automatic estimation method of word weights based on the influences on IR.

2 Evaluation Measure of ASR for IR

2.1 Weighted Word Error Rate (WWER)

The conventional ASR evaluation measure, namely, word error rate (WER), is defined as Equation (1).

$$\text{WER} = \frac{I + D + S}{N} \tag{1}$$

Here, *N* is the number of words in the correct transcript, *I* is the number of incorrectly inserted words, *D* is the number of deletion errors, and *S* is the number of substitution errors. For each utterance, DP matching of the ASR result and the correct transcript is performed to identify the correct words and calculate WER.

Apparently in WER, all words are treated uniformly or with the same weight. However, there must be a difference in the weight of errors, since several keywords have more impact on IR or the understanding of the speech than trivial functional words. Based on the background, WER is generalize and weighted WER (WWER), in which each word has a different weight that reflects its influence on IR, is introduced. WWER is defined as follows.

ASR result	:	a	b	c	d	e	f	
Correct transcript	:	a		c	d'		f	g
DP result	:	*C*	*I*	*C*	*S*		*C*	*D*

$$\text{WWER} = (V_I + V_D + V_S)/V_N$$

$$V_N = v_a + v_c + v_{d'} + v_f + v_g,\ V_I = v_b$$
$$V_D = v_g,\ V_S = \max(v_d + v_e, v_{d'})$$

v_i: weight of word *i*

Figure 1: Example of WWER calculation

$$\text{WWER} = \frac{V_I + V_D + V_S}{V_N} \tag{2}$$

$$V_N = \Sigma_{w_i} \quad v_{w_i} \tag{3}$$

$$V_I = \Sigma_{\hat{w}_i \in I} \quad v_{\hat{w}_i} \tag{4}$$

$$V_D = \Sigma_{w_i \in D} \quad v_{w_i} \tag{5}$$

$$V_S = \Sigma_{seg_j \in S} \quad v_{seg_j} \tag{6}$$

$$v_{seg_j} = \max(\Sigma_{\hat{w}_i \in seg_j} v_{\hat{w}_i}, \Sigma_{w_i \in seg_j} v_{w_i})$$

Here, v_{w_i} is the weight of word w_i, which is the i-th word of the correct transcript, and $v_{\hat{w}_i}$ is the weight of word $\hat{w}_i$, which is the i-th word of the ASR result. seg_j represents the j-th substituted segment, and v_{seg_j} is the weight of segment seg_j. For segment seg_j, the total weight of the correct words and the recognized words are calculated, and then the larger one is used as v_{seg_j}. In this work, we use alignment for WER to identify the correct words and calculate WWER. Thus, WWER equals WER if all word weights are set to 1. In Fig. 1, an example of a WWER calculation is shown.

WWER calculated based on ideal word weights represents IR performance degradation when the ASR result is used as a query for IR. Thus, we must perform ASR to minimize WWER for speech-based IR.

2.2 Minimum Bayes-Risk Decoding

Next, a decoding strategy to minimize WWER based on the Minimum Bayes-Risk framework (V.Goel et al., 1998) is described.

In Bayesian decision theory, ASR is described with a decision rule $\delta(X)$: $X \rightarrow \hat{W}$. Using a real-valued loss function $l(W, \delta(X)) = l(W, W')$, the

decision rule minimizing Bayes-risk is given as follows. It is equivalent to the orthodox ASR (maximum likelihood ASR) when a 0/1 loss function is used.

$$\delta(X) = \underset{W}{\text{argmin}} \sum_{W'} l(W, W') \cdot P(W'|X) \quad (7)$$

The minimization of WWER is realized using WWER as a loss function (H.Nanjo and T.Kawahara, 2005) (H.Nanjo et al., 2005).

3 Estimation of Word Weights

A word weight should be defined based on its influence on IR. Specifically, weights are estimated so that WWER will be equivalent to an IR performance degradation. For an evaluation measure of IR performance degradation, IR score degradation ratio (IRDR), which is described in detail in Section 4.2, is introduced in this work. The estimation of weights is performed as follows.

1. Query pairs of a spoken-query recognition result and its correct transcript are set as training data. For each query pair m, do procedures 2 to 5.
2. Perform IR with a correct transcript and calculate IR score R_m.
3. Perform IR with a spoken-query ASR result and calculate IR score H_m.
4. Calculate IR score degradation ratio (IRDR$_m = 1 - \frac{H_m}{R_m}$).
5. Calculate WWER$_m$.
6. Estimate word weights so that WWER$_m$ and IRDR$_m$ are equivalent for all queries.

Practically, procedure 6 is defined to minimize the mean square error between both evaluation measures (WWER and IRDR) as follows.

$$F(\mathbf{x}) = \sum_m \left(\frac{E_m(\mathbf{x})}{C_m(\mathbf{x})} - \text{IRDR}_m \right)^2 \rightarrow \min \quad (8)$$

Here, $\mathbf{x}$ is a vector that consists of the weights of words. $E_m(\mathbf{x})$ is a function that determines the sum of the weights of mis-recognized words. $C_m(\mathbf{x})$ is a function that determines the sum of the weights of the correct transcript. $E_m(\mathbf{x})$ and $C_m(\mathbf{x})$ correspond to the numerator and denominator of Equation (2), respectively.

In this work, we adopt the steepest decent method to determine the weights that give minimal $F(\mathbf{x})$. Initially, all weights are set to 1, and then each word weight (x_k) is iteratively updated based on Equation (9) until the mean square error between WWER and IRDR is converged.

$$x_k{}' = \begin{cases} x_k - \alpha & \text{if } \dfrac{\partial F}{\partial x_k} > 0 \\ x_k + \alpha & \text{else if } \dfrac{\partial F}{\partial x_k} < 0 \\ x_k & \text{otherwise} \end{cases} \quad (9)$$

where

$$\begin{aligned} \frac{\partial F}{\partial x_k} &= \sum_m 2 \left(\frac{E_m}{C_m} - \text{IRDR}_m \right) \cdot \left(\frac{E_m}{C_m} - \text{IRDR}_m \right)' \\ &= \sum_m 2 \left(\frac{E_m}{C_m} - \text{IRDR}_m \right) \cdot \frac{E_m' \cdot C_m - E_m \cdot C_m'}{C_m^2} \\ &= \sum_m 2 \left(\frac{E_m}{C_m} - \text{IRDR}_m \right) \cdot \frac{1}{C_m} \left(E_m' - C_m' \cdot \frac{E_m}{C_m} \right) \\ &= \sum_m \frac{2}{C_m} \left(\text{WWER}_m - \text{IRDR}_m \right) \left(E_m' - C_m' \cdot \text{WWER}_m \right) \end{aligned}$$

4 Weight Estimation on Orthodox IR

4.1 WEB Page Retrieval

In this paper, weight estimation is evaluated with an orthodox IR system that searches for appropriate documents using statistical matching for a given query. The similarity between a query and documents is defined by the inner product of the feature vectors of the query and the specific document. In this work, a feature vector that consists of TF-IDF values is used. The TF-IDF value is calculated for each word t and document (query) i as follows.

$$\text{TF-IDF}(t, i) = \frac{tf_{t,i}}{\frac{\text{DL}_i}{\text{avglen}} + tf_{t,i}} \cdot \log \frac{N}{df_t} \quad (10)$$

Here, term frequency $tf_{t,i}$ represents the occurrence counts of word t in a specific document i, and document frequency df_t represents the total number

of documents that contain word t. A word that occurs frequently in a specific document and rarely occurs in other documents has a large TF-IDF value. We normalize TF values using length of the document (DL_i) and average document lengths over all documents (avglen) because longer document have more words and TF values tend to be larger.

For evaluation data, web retrieval task "NTCIR-3 WEB task", which is distributed by NTCIR (NTC,), is used. The data include web pages to be searched, queries, and answer sets. For speech-based information retrieval, 470 query utterances by 10 speakers are also included.

4.2 Evaluation Measure of IR

For an evaluation measure of IR, discount cumulative gain (DCG) is used, and described below.

$$\mathrm{DCG}(i) = \begin{cases} g(1) & \text{if } i = 1 \\ \mathrm{DCG}(i-1) + \dfrac{g(i)}{\log(i)} & otherwise \end{cases} \tag{11}$$

$$g(i) = \begin{cases} h & \text{if } d_i \in H \\ a & \text{else if } d_i \in A \\ b & \text{else if } d_i \in B \\ c & \text{otherwise} \end{cases}$$

Here, d_i represents i-th retrieval result (document). H, A, and B represent a degree of relevance; H is labeled to documents that are highly relevant to the query. A and B are labeled to documents that are relevant and partially relevant to the query, respectively. "h", "a", "b", and "c" are the gains, and in this work, $(h, a, b, c) = (3, 2, 1, 0)$ is adopted. When retrieved documents include many relevant documents that are ranked higher, the DCG score increases.

In this work, word weights are estimated so that WWER and IR performance degradation will be equivalent. For an evaluation measure of IR performance degradation, we define IR score degradation ratio (IRDR) as below.

$$\mathrm{IRDR} = 1 - \frac{H}{R} \tag{12}$$

R represents a DCG score calculated with IR results by text query, and H represents a DCG score given by the ASR result of the spoken query. IRDR represents the ratio of DCG score degradation affected by ASR errors.

4.3 Automatic speech recognition system

In this paper, ASR system is set up with following acoustic model, language model and a decoder Julius rev.3.4.2(A.Lee et al., 2001). As for acoustic model, gender independent monophone model (129 states, 16 mixtures) trained with JNAS corpus are used. Speech analysis is performed every 10 msec. and a 25 dimensional parameter is computed (12 MFCC + 12ΔMFCC + ΔPower). For language model, a word trigram model with the vocabulary of 60K words trained with WEB text is used.

Generally, trigram model is used as acoustic model in order to improve the recognition accuracy. However, monophone model is used in this paper, since the proposed estimation method needs recognition error (and IRDR).

4.4 Results

4.4.1 Correlation between Conventional ASR and IR Evaluation Measures

We analyzed the correlations of conventional ASR evaluation measures with IRDR by selecting appropriate test data as follows. First, ASR is performed for 470 spoken queries of an NTCIR-3 web task. Then, queries are eliminated whose ASR results do not contain recognition errors and queries with which no IR results are retrieved. Finally, we selected 107 pairs of query transcripts and their ASR results as test data.

For all 107 pairs, we calculated WER and IRDR using corresponding ASR result. Figure 2 shows the correlations between WER and IRDR. Correlation coefficient between both is 0.119. WER is not correlated with IRDR. Since our IR system only uses the statistics of nouns, WER is not an appropriate evaluation measure for IR. Conventionally, for such tasks, keyword recognition has been performed, and keyword error rate (KER) has been used as an evaluation measure. KER is calculated by setting all keyword weights to 1 and all weights of the other words to 0 in WWER calculation. Figure 3 shows the correlations between KER and IRDR. Although IRDR is more correlated with KER than WER, KER is not significantly correlated with IRDR (correlation coefficient: 0.224). Thus, KER is not a suitable evaluation measure of ASR for IR. This fact shows that each keyword has a different influence on IR and

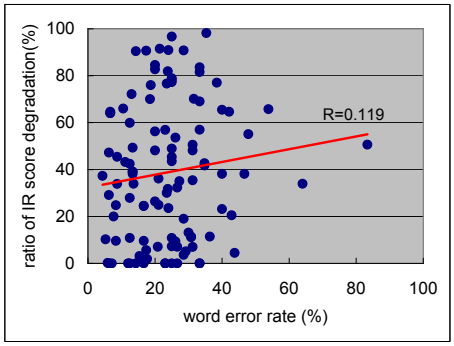


Figure 2: Correlation between ratio of IR score degradation and WER

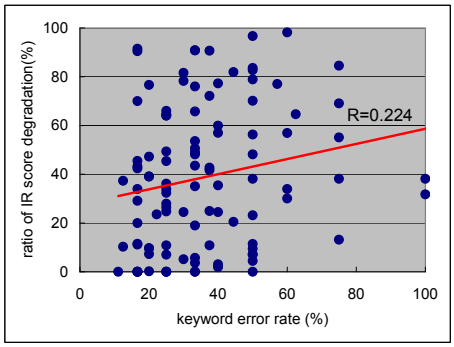


Figure 3: Correlation between ratio of IR score degradation and KER

should be given a different weight based on its influence on IR.

4.4.2 Correlation between WWER and IR Evaluation Measure

In ASR for IR, since some words are significant, each word should have a different weight. Thus, we assume that each keyword has a positive weight, and non-keywords have zero weight. WWER calculated with these assumptions is then defined as weighted keyword error rate (WKER).

Using the same test data (107 queries), keyword weights were estimated with the proposed estimation method. The correlation between IRDR and WKER calculated with the estimated word weights is shown in Figure 4. A high correlation between IRDR and WKER is confirmed (correlation coefficient: 0.969). The result shows that the proposed method works well and proves that giving a different weight to each word is significant.

The proposed method enables us to extend text-based IR systems to speech-based IR systems with typical text queries for the IR system, ASR results of the queries, and answer sets for each query. ASR results are not necessary since they can be substituted with simulated texts that can be automatically generated by replacing some words with others. On the contrary, text queries and answer sets are indispensable and must be prepared. It costs too much to make answer sets manually since we should consider whether each answer is relevant to the query. For these reasons, it is difficult to apply the method to a large-scale speech-based IR system. An estimation method without hand-labeled answer sets is strongly required.

An estimation method without hand-labeled answer sets, namely, the unsupervised estimation of word weights, is also tested. Unsupervised estimation is performed as described in Section 3. In unsupervised estimation, the IR result (document set) with a correct transcript is regarded as an answer set, namely, a presumed answer set, and it is used for IRDR calculation instead of a hand-labeled answer set.

The result (correlation between IRDR and WKER) is shown in Figure 5. Without hand-labeled answer sets, we obtained high correlation (0.712 of correlation coefficient) between IRDR and WKER. The result shows that the proposed estimation method is effective and widely applicable to IR systems since it requires only typical text queries for IR. With the WWER given by the estimated weights, IR performance degradation can be confidently predicted. It is confirmed that the ASR approach to minimize such WWER, which is realized with decoding based on a Minimum Bayes-Risk framework (H.Nanjo and T.Kawahara, 2005)(H.Nanjo et al., 2005), is effective for IR.

4.5 Discussion

In this section, we discuss the problem of word weight estimation. Although we obtained high correlation between IRDR and KWER, the estimation may encounter the over-fitting problem when we use small estimation data. When we want to design a speech-based IR system, a sufficient size of typical queries is often prepared, and thus, our proposed method can estimate appropriate weights for typical significant words. Moreover, this problem will be

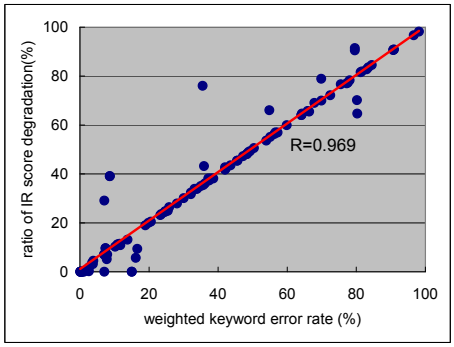


Figure 4: Correlation between ratio of IR score degradation and WKER (supervised estimation)

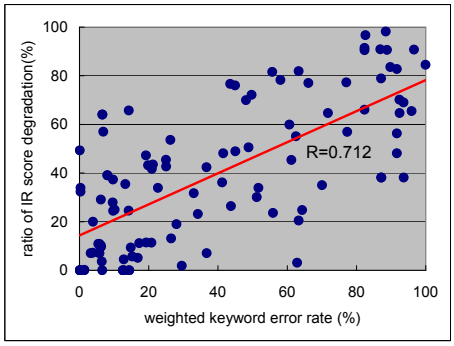


Figure 5: Correlation between ratio of IR score degradation and WKER (unsupervised estimation)

avoided using a large amount of dummy data (pair of query and IRDR) with unsupervised estimation. In this work, although obtained correlation coefficient of 0.712 in unsupervised estimation, it is desirable to obtain much higher correlation. There are much room to improve unsupervised estimation method.

In addition, since typical queries for IR system will change according to the users, current topic, and so on, word weights should be updated accordingly. It is reasonable approach to update word weights with small training data which has been input to the system currently. For such update system, our estimation method, which may encounter the over-fitting problem to the small training data, may work as like as cache model (P.Clarkson and A.J.Robinson, 1997), which gives higher language model probability to currently observed words.

5 Conclusion

We described the automatic estimation of word significance for IR-oriented ASR. The proposed estimation method only requires typical queries for the IR, and estimates weights of words so that WWER, which is an evaluation measure for ASR, will be equivalent to IRDR, which represents a degree of IR degradation when an ASR result is used as a query for IR. The proposed estimation method was evaluated on a web page retrieval task. WWER based on estimated weights is highly correlated with IRDR. It is confirmed that the proposed method is effective and we can predict IR performance confidently with such WWER, which shows the effectiveness of our proposed ASR approach minimizing such WWER for IR.

Acknowledgment: The work was partly supported by KAKENHI WAKATE(B).

References

A.Lee, T.Kawahara, and K.Shikano. 2001. Julius – an open source real-time large vocabulary recognition engine. In *Proc. EUROSPEECH*, pages 1691–1694.

C.Hori, T.Hori, H.Isozaki, E.Maeda, S.Katagiri, and S.Furui. 2003. Deriving disambiguous queries in a spoken interactive ODQA system. In *Proc. IEEE-ICASSP*, pages 624–627.

E.Levin, S.Narayanan, R.Pieraccini, K.Biatov, E.Bocchieri, G.D.Fabbrizio, W.Eckert, S.Lee, A.Pokrovsky, M.Rahim, P.Ruscitti, and M.Walker. 2000. The AT&T-DARPA communicator mixed-initiative spoken dialogue system. In *Proc. ICSLP*.

H.Nanjo and T.Kawahara. 2005. A new ASR evaluation measure and minimum Bayes-risk decoding for open-domain speech understanding. In *Proc. IEEE-ICASSP*, pages 1053–1056.

H.Nanjo, T.Misu, and T.Kawahara. 2005. Minimum Bayes-risk decoding considering word significance for information retrieval system. In *Proc. INTERSPEECH*, pages 561–564.

NTCIR project web page. `http://research.nii.ac.jp/ntcir/`.

P.Clarkson and A.J.Robinson. 1997. Language Model Adaptation using Mixtures and an Exponentially Decaying cache. In *Proc. IEEE-ICASSP*, volume 2, pages 799–802.

T.Misu, K.Komatani, and T.Kawahara. 2004. Confirmation strategy for document retrieval systems with spoken dialog interface. In *Proc. ICSLP*, pages 45–48.

V.Goel, W.Byrne, and S.Khudanpur. 1998. LVCSR rescoring with modified loss functions: A decision theoretic perspective. In *Proc. IEEE-ICASSP*, volume 1, pages 425–428.

Rapid Prototyping of Robust Language Understanding Modules for Spoken Dialogue Systems

[†]Yuichiro Fukubayashi, [†]Kazunori Komatani, [‡]Mikio Nakano,
[‡]Kotaro Funakoshi, [‡]Hiroshi Tsujino, [†]Tetsuya Ogata, [†]Hiroshi G. Okuno

[†]Graduate School of Informatics, Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto
606-8501, Japan
{fukubaya,komatani}@kuis.kyoto-u.ac.jp
{ogata,okuno}@kuis.kyoto-u.ac.jp

[‡]Honda Research Institute Japan Co., Ltd.
8-1 Honcho, Wako, Saitama
351-0188, Japan
nakano@jp.honda-ri.com
{funakoshi,tsujino}@jp.honda-ri.com

Abstract

Language understanding (LU) modules for spoken dialogue systems in the early phases of their development need to be (i) easy to construct and (ii) robust against various expressions. Conventional methods of LU are not suitable for new domains, because they take a great deal of effort to make rules or transcribe and annotate a sufficient corpus for training. In our method, the weightings of the Weighted Finite State Transducer (WFST) are designed on two levels and simpler than those for conventional WFST-based methods. Therefore, our method needs much fewer training data, which enables rapid prototyping of LU modules. We evaluated our method in two different domains. The results revealed that our method outperformed baseline methods with less than one hundred utterances as training data, which can be reasonably prepared for new domains. This shows that our method is appropriate for rapid prototyping of LU modules.

1 Introduction

The language understanding (LU) of spoken dialogue systems in the early phases of their development should be trained with a small amount of data in their construction. This is because large amounts of annotated data are not available in the early phases. It takes a great deal of effort and time to transcribe and provide correct LU results to a large amount of data. The LU should also be robust, i.e., it should be accurate even if some automatic speech recognition (ASR) errors are contained in its input. A robust LU module is also helpful when collecting dialogue data for the system because it suppresses incorrect LU and unwanted behaviors. We developed a method of rapidly prototyping LU modules that is easy to construct and robust against various expressions. It makes LU modules in the early phases easier to develop.

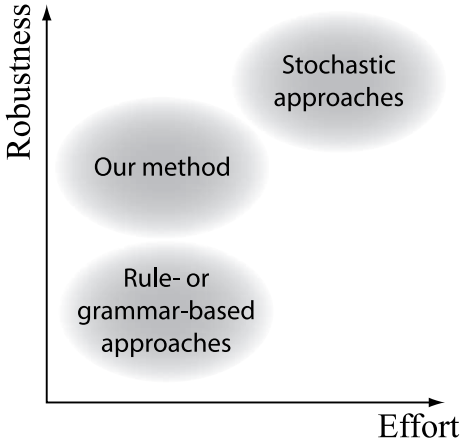


Figure 1: Relationship between our method and conventional methods

Several methods of implementing an LU module in spoken dialogue systems have been proposed. Using grammar-based ASR is one of the simplest. Although its ASR output can easily be transformed into concepts based on grammar rules, complicated grammars are required to understand the user's utterances in various expressions. It takes a great deal of effort to the system developer. Extracting con-

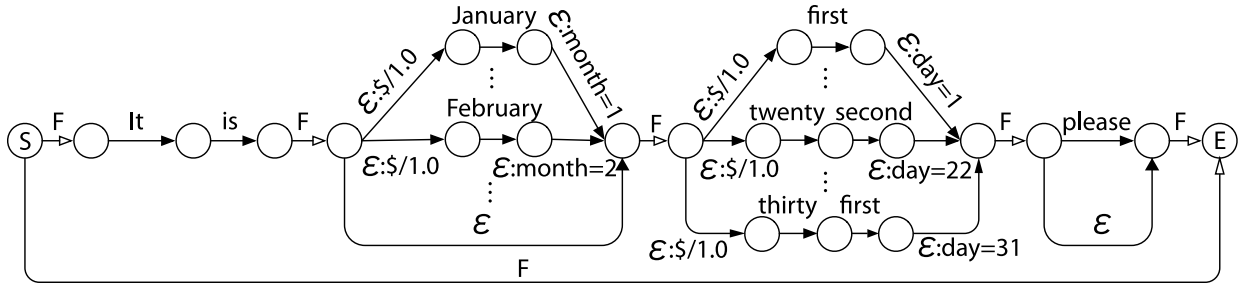


Figure 2: Example of WFST for LU

cepts from user utterances by keyword spotting or heuristic rules has also been proposed (Seneff, 1992) where utterances can be transformed into concepts without major modifications to the rules. However, numerous complicated rules similarly need to be manually prepared. Unfortunately, neither method is robust against ASR errors.

To cope with these problems, corpus-based (Sudoh and Tsukada, 2005; He and Young, 2005) and Weighted Finite State Transducer (WFST)-based methods (Potamianos and Kuo, 2000; Wutiwiwatchai and Furui, 2004) have been proposed as LU modules for spoken dialogue systems. Since these methods extract concepts using stochastic analysis, they do not need numerous complicated rules. These, however, require a great deal of training data to implement the module and are not suitable for constructing new domains.

Here, we present a new WFST-based LU module that has two main features.

1. A statistical language model (SLM) for ASR and a WFST for parsing that are automatically generated from the domain grammar description.

2. Since the weighting for the WFST is simpler than that in conventional methods, it requires fewer training data than conventional weighting schemes.

Our method accomplishes robust LU with less effort using SLM-based ASR and WFST parsing. Figure 1 outlines the relationships between our method and conventional schemes. Since rule- or grammar-based approaches do not require a large amount of data, they take less effort than stochastic techniques. However, they are not robust against ASR errors. Stochastic approaches, on the contrary, take a great deal of effort to collect data but are robust against ASR errors. Our method is an intermediate approach that lies between these. That is, it is more robust than rule- or grammar-based approaches and takes less effort than stochastic techniques. This characteristic makes it easier to rapidly prototype LU modules for a new domain and helps development in the early phases.

2 Related Work and WFST-based Approach

A Finite State Transducer (FST)-based LU is explained here, which accepts ASR output as its input. Figure 2 shows an example of the FST for a video recording reservation domain. The input, ε, means that a transition with no input is permitted at the state transition. In this example, the LU module returns the concept [month=2, day=22] for the utterance "It is February twenty second please". Here, a FILLER transition in which any word is accepted is appropriately allowed between phrases. In Figure 2, 'F' represents 0 or more FILLER transitions. A FILLER transition from the start to the end is inserted to reject unreliable utterances. This FILLER transition enables us to ignore unnecessary words listed in the example utterances in Table 1. The FILLER transition helps to suppress the insertion of incorrect concepts into LU results.

However, many output sequences are obtained for one utterance due to the FILLER transitions, because the utterance can be parsed with several paths. We used a WFST to select the most appropriate path from several output sequences. The path with the highest cumulative weight, w, is selected in a

Table 2: Many LU results for input "It is February twenty second please"

LU output					LU result	w
It	is	February	twenty second	please	month=2, day=22	2.0
It	is	FILLER	twenty second	please	day=22	1.0
It	is	FILLER	twenty second	FILLER	day=22	1.0
FILLER	FILLER	FILLER	FILLER FILLER	FILLER	n/a	0

Table 1: Examples of utterances with FILLERs

ASR output
Well, it is February twenty second please
It is **uhm,** February twenty second please
It is February, **twe-**, twenty second please
It is February twenty second please, **OK?**

(LU result = [month=2, day=22])

WFST-based LU. In the example in Table 2, the concept [month=2, day=22] has been selected, because its cumulative weight, w, is 2.0, which is the highest.

The weightings of conventional WFST-based approaches used an n-gram of concepts (Potamianos and Kuo, 2000) and that of word-concept pairs (Wutiwiwatchai and Furui, 2004). They obtained the n-grams from several thousands of annotated utterances. However, it takes a great deal of effort to transcribe and annotate a large corpus. Our method enables prototype LU modules to be rapidly constructed that are robust against various expressions with SLM-based ASR and WFST-based parsing. The SLM and WFST are generated automatically from a domain grammar description in our toolkit. We need fewer data to train WFST, because its weightings are simpler than those in conventional methods. Therefore, it is easy to develop an LU module for a new domain with our method.

3 Domain Grammar Description

A developer defines grammars, slots, and concepts in a domain in an XML file. This description enables an SLM for ASR and parsing WFST to be automatically generated. Therefore, a developer can construct an LU module rapidly with our method.

Figure 3 shows an example of a description. A definition of a slot is described in `keyphrase-class` tags and its keyphrases and

```
...
<keyphrase-class name="month">
...
  <keyphrase>
    <orth>February</orth>
    <sem>2</sem>
  </keyphrase>
...
</keyphrase-class>
...
<action type="specify-attribute">
  <sentence> {It is} [*month] *day [please]
  </sentence>
</action>
```

Figure 3: Example of a grammar description

the values are in `keyphrase` tags. The `month` is defined as a slot in this figure. `February` and `2` are defined as one of the phrases and values for the slot `month`. A grammar is described in a sequence of terminal and non-terminal symbols. A non-terminal symbol represents a class of keyphrases, which is defined in `keyphrase-class`. It begins with an asterisk "`*`" in a grammar description in `sentence` tags. Symbols that can be skipped are enclosed by brackets `[ ]`. The FILLER transition described in Section 2 is inserted between the symbols unless they are enclosed in brackets `[ ]` or braces `{ }`. Braces are used to avoid FILLER transitions from being inserted. For example, the grammar in Figure 3 accepts "It is February twenty second please." and "It is twenty second, OK?", but rejects "It is February." and "It, uhm, is February twenty second.".

A WFST for parsing can be automatically generated from this XML file. The WFST in Figure 2 is generated from the definition in Figure 3. Moreover, we can generate example sentences from the grammar description. The SLM for the speech recognizer is generated with our method by using many example sentences generated from the defined grammar.

4 Weighting for ASR Outputs on Two Levels

We define weights on two levels for a WFST. The first is *a weighting for ASR outputs*, which is set to select paths that are reliable at a surface word level. The second is *a weighting for concepts*, which is used to select paths that are reliable on a concept level. The weighting for concepts reflects correctness at a more abstract level than the surface word level. The weighting for ASR outputs consists of two categories: a weighting for ASR N-best outputs and one for accepted words. We will describe the definitions of these weightings in the following subsections.

4.1 Weighting for ASR N-Best Outputs

The N-best outputs of ASR are used for an input of a WFST. Weights are assigned to each sentence in ASR N-best outputs. Larger weights are given to more reliable sentences, whose ranks in ASR N-best are higher. We define this preference as

$$w_s^i = \frac{e^{\beta \cdot score_i}}{\sum_j^N e^{\beta \cdot score_j}},$$

where w_s^i is a weight for the i-th sentence in ASR N-best outputs, β is a coefficient for smoothing, and $score_i$ is the log-scaled score of the i-th ASR output. This weighting reflects the reliability of the ASR output. We set β to 0.025 in this study after a preliminary experiment.

4.2 Weighting for Accepted Words

Weights are assigned to word sequences that have been accepted by the WFST. Larger weights are given to more reliable sequences of ASR outputs at the surface word level. Generally, longer sequences having more words that are not fillers and more reliable ASR outputs are preferred. We define these preferences as the weights:

1. **word(const.)**: $w_w = 1.0$,
2. **word(#phone)**: $w_w = l(W)$, and
3. **word(CM)**: $w_w = CM(W) - \theta_w$.

The **word(const.)** gives a constant weight to all accepted words. This means that sequences with more words are simply preferred. The **word(#phone)** takes the length of each accepted word into consideration. This length is measured by its number of phonemes, which are normalized by that of the longest word in the vocabulary. The normalized values are denoted as $l(W)$ ($0 < l(W) \leq 1$). By adopting **word(#phone)**, the length of sequences is represented more accurately. We also take the reliability of the accepted words into account as **word(CM)**. This uses confidence measures (Lee et al., 2004) for a word, W, in ASR outputs, which are denoted as $CM(W)$. The θ_w is the threshold for determining whether word W is accepted or not. The w_w obtains a negative value for an unreliable word W when $CM(W)$ is lower than θ_w. This represents a preference for longer and more reliable sequences.

4.3 Weighting for Concepts

In addition to the ASR level, weights on a concept level are also assigned. The concepts are obtained from the parsing results by the WFST, and contain several words. Weights for concepts are defined by using the measures of all words contained in a concept.

We prepared three kinds of weights for the concepts:

1. **cpt(const.)**: $w_c = 1.0$,
2. **cpt(avg)**:
$$w_c = \frac{\sum_{\boldsymbol{W}}(CM(W) - \theta_c)}{\#\boldsymbol{W}}, \text{and}$$
3. **cpt(#pCM(avg))**:
$$w_c = \frac{\sum_{\boldsymbol{W}}(CM(W) \cdot l(W) - \theta_c)}{\#\boldsymbol{W}},$$

where $\boldsymbol{W}$ is a set of accepted words, W, in the corresponding concept, and $\#\boldsymbol{W}$ is the number of words in $\boldsymbol{W}$.

The **cpt(const.)** represents a preference for sequences with more concepts. The **cpt(avg)** is defined as the weight by using the $CM(W)$ of each word contained in the concept. The **cpt(#pCM(avg))** represents a preference for longer and reliable sequences with more concepts. The θ_c is the threshold for the acceptance of a concept.

Table 3: Examples of weightings when parameter set is: **word(CM)** and **cpt(#pCM(avg))**

ASR onput	No,	it	is	February	twenty	second
LU output	FILLER	it	is	February	twenty	second
$CM(W)$	0.3	0.7	0.6	0.9	1.0	0.9
$l(W)$	0.3	0.2	0.2	0.9	0.6	0.5
Concept	-	-	-	month=2	day=22	
word	-	$0.7-\theta_w$	$0.6-\theta_w$	$0.9-\theta_w$	$1.0-\theta_w$	$0.9-\theta_w$
cpt	-	-	-	$(0.9\cdot 0.9-\theta_c)/1$	$(1.0\cdot 0.6-\theta_c+0.9\cdot 0.5-\theta_c)/2$	

Reference	From	June	third			please	
ASR output	From	June	third	uhm	FIT	please	LU result
$CM(W)$	0.771	0.978	0.757	0.152	0.525	0.741	
LU reference	From	June	third	FILLER	FILLER	FILLER	month:6, day:3
Our method	From	June	third	FILLER	FILLER	FILLER	month:6, day:3
Keyword spotting	From	June	third	FILLER	FIT	please	month:6, day:3, car:FIT

('FIT' is the name of a car.)

Figure 4: Example of LU with WFST

4.4 Calculating Cumulative Weight and Training

The LU results are selected based on the weighted sum of the three weights in Subsection 4.3 as

$$w^i = w_s^i + \alpha_w \sum w_w + \alpha_c \sum w_c$$

The LU module selects an output sequence with the highest cumulative weight, w^i, for $1 \leq i \leq N$.

Let us explain how to calculate cumulative weight w^i by using the example specified in Table 3. Here, **word(CM)** and **cpt(#pCM(avg))** are selected as parameters. The sum of weights in this table for accepted words is $\alpha_w(4.1-5\theta_w)$, when the input sequence is "No, it is February twenty second.". The sum of weights for concepts is $\alpha_c(1.335-2\theta_c)$ because the weight for "month=2" is $\alpha_c(0.81-\theta_c)$ and the weight for "day=22" is $\alpha_c(0.525-\theta_c)$. Therefore, cumulative weight w^i for this input sequence is $w_s^i + \alpha_w(4.1-5\theta_w) + \alpha_c(1.335-2\theta_c)$.

In the training phase, various combinations of parameters are tested, i.e., which weightings are used for each of ASR output and concept level, such as $N = 1$ or 10, coefficient $\alpha_{w,c} = 1.0$ or 0, and threshold $\theta_{w,c} = 0$ to 0.9 at intervals of 0.1, on the training data. The coefficient $\alpha_{w,c} = 0$ means that a corresponding weight is not added. The optimal parameter settings are obtained after testing the various combinations of parameters. They make the concept error rate (CER) minimum for a training data set. We calculated the CER in the following equation: $CER = (S+D+I)/N$, where N is the number of concepts in a reference, and S, D, and I correspond to the number of substitution, deletion, and insertion errors.

Figure 4 shows an example of LU with our method, where it rejects misrecognized concept [car:FIT], which cannot be rejected by keyword spotting.

5 Experiments and Evaluation

5.1 Experimental Conditions

We discussed our experimental investigation into the effects of weightings in Section 4. The user utterance in our experiment was first recognized by ASR. Then, the i-th sentence of ASR output was input to WFST for $1 \leq i \leq N$, and the LU result for the highest cumulative weight, w^i, was obtained.

We used 4186 utterances in the video recording reservation domain (video domain), which consisted of eight different dialogues with a total of 25 different speakers. We also used 3364 utterances in the rent-a-car reservation domain (rent-a-car domain) of

eight different dialogues with 23 different speakers. We used Julius [1] as a speech recognizer with an SLM. The language model was prepared by using example sentences generated from the grammars of both domains. We used 10000 example sentences in the video and 40000 in the rent-a-car domain. The number of the generated sentences was determined empirically. The vocabulary size was 209 in the video and 891 in the rent-a-car domain. The average ASR accuracy was 83.9% in the video and 65.7% in the rent-a-car domain. The grammar in the video domain included phrases for dates, times, channels, commands. That of the rent-a-car domain included phrases for dates, times, locations, car classes, options, and commands. The WFST parsing module was implemented by using the MIT FST toolkit (Hetherington, 2004).

5.2 Performance of WFST-based LU

We evaluated our method in the two domains: video and rent-a-car. We compared the CER on test data, which was calculated by using the optimal settings for both domains. We evaluated the results with 4-fold cross validation. The number of utterances for training was 3139 (=4186*(3/4)) in the video and 2523 (=3364*(3/4)) in the rent-a-car domain.

The baseline method was simple keyword spotting because we assumed a condition where a large amount of training data was not available. This method extracts as many keyphrases as possible from ASR output without taking speech recognition errors and grammatical rules into consideration. Both grammar-based and SLM-based ASR outputs are used for input in keyword spotting (denoted as "Grammar & spotting" and "SLM & spotting" in Table 4). The grammar for grammar-based ASR was automatically generated by the domain description file. The accuracy of grammar-based ASR was 66.3% in the video and 43.2% in the rent-a-car domain.

Table 4 lists the CERs for both methods. In keyword spotting with SLM-based ASR, the CERs were improved by 5.2 points in the video and by 22.2 points in the rent-a-car domain compared with those with grammar-based ASR. This is because SLM-based ASR is more robust against fillers and unknown words than grammar-based ASR. The CER was improved by 3.4 and 6.9 points by optimal weightings for WFST. Table 5 lists the optimal parameters in both domains. The $\alpha_c = 0$ in the video domain means that weights for concepts were not used. This result shows that optimal parameters depend on the domain for the system, and these need to be adapted for each domain.

[1] http://julius.sourceforge.jp/

Table 4: Concept error rates (CERs) in each domain

Domain	Grammar & spotting	SLM & spotting	Our method
Video	22.1	16.9	13.5
Rent-a-car	51.1	28.9	22.0

5.3 Performance According to Training Data

We also investigated the relationship between the size of the training data for our method and the CER. In this experiment, we calculated the CER in the test data by increasing the number of utterances for training. We also evaluated the results by 4-fold cross validation.

Figures 5 and 6 show that our method outperformed the baseline methods by about 80 utterances in the video domain and about 30 utterances in the rent-a-car domain. These results mean that our method can effectively be used to rapidly prototype LU modules. This is because it can achieve robust LU with fewer training data compared with conventional WFST-based methods, which need over several thousand sentences for training.

6 Conclusion

We developed a method of rapidly prototyping robust LU modules for spoken language understanding. An SLM for a speech recognizer and a WFST for parsing were automatically generated from a domain grammar description. We defined two kinds of weightings for the WFST at the word and concept levels. These two kinds of weightings were calculated by ASR outputs. This made it possible to create an LU module for a new domain with less effort because the weighting scheme was relatively simpler than those of conventional methods. The optimal parameters could be selected with fewer training data in both domains. Our experiment re-

Table 5: Optimal parameters in each domain

Domain	N	α_w	w_w	α_c	w_c
Video	1	1.0	**word(const.)**	0	-
Rent-a-car	10	1.0	**word(CM)**-0.0	1.0	**cpt(#pCM(avg))**-0.8

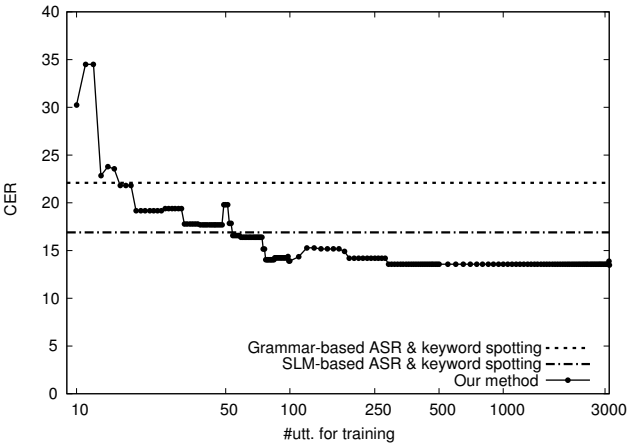


Figure 5: CER in video domain

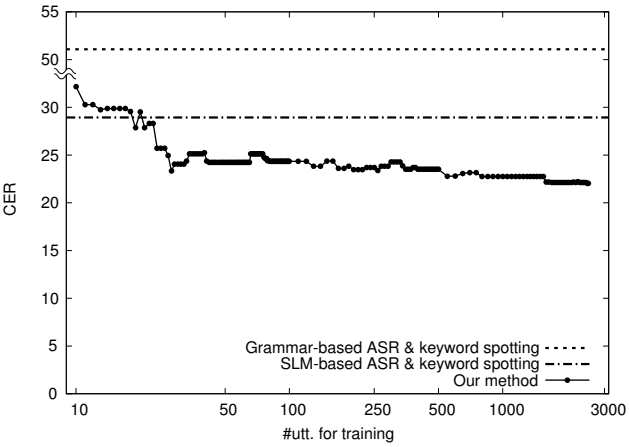


Figure 6: CER in rent-a-car domain

vealed that the CER could be improved compared to the baseline by training optimal parameters with a small amount of training data, which could be reasonably prepared for new domains. This means that our method is appropriate for rapidly prototyping LU modules. Our method should help developers of spoken dialogue systems in the early phases of development. We intend to evaluate our method on other domains, such as database searches and question answering in future work.

Acknowledgments

We are grateful to Dr. Toshihiko Ito and Ms. Yuka Nagano of Hokkaido University for constructing the rent-a-car domain system.

References

Yulan He and Steve Young. 2005. Spoken Language Understanding using the Hidden Vector State Model. *Speech Communication*, 48(3-4):262–275.

Lee Hetherington. 2004. The MIT finite-state transducer toolkit for speech and language processing. In *Proc. 6th International Conference on Spoken Language Processing (INTERSPEECH-2004 ICSLP)*.

Akinobu Lee, Kiyohiro Shikano, and Tatsuya Kawahara. 2004. Real-time word confidence scoring using local posterior probabilities on tree trellis search. In *Proc. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, volume 1, pages 793–796.

Alexandors Potamianos and Hong-Kwang J. Kuo. 2000. Statistical recursive finite state machine parsing for speech understanding. In *Proc. 6th International Conference on Spoken Language Processing (INTERSPEECH-2000 ICSLP)*, pages 510–513.

Stephanie Seneff. 1992. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86.

Katsuhito Sudoh and Hajime Tsukada. 2005. Tightly integrated spoken language understanding using word-to-concept translation. In *Proc. 9th European Conference on Speech Communication and Technology (INTERSPEECH-2005 Eurospeech)*, pages 429–432.

Chai Wutiwiwatchai and Sadaoki Furui. 2004. Hybrid statistical and structural semantic modeling for Thai multi-stage spoken language understanding. In *Proc. HLT-NAACL Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Information for Speech Processing*, pages 2–9.

Automatic Prosodic Labeling with Conditional Random Fields and Rich Acoustic Features

Gina-Anne Levow
University of Chicago
Department of Computer Science
1100 E. 58th St.
Chicago, IL 60637 USA
levow@cs.uchicago.edu

Abstract

Many acoustic approaches to prosodic labeling in English have employed only local classifiers, although text-based classification has employed some sequential models. In this paper we employ linear chain and factorial conditional random fields (CRFs) in conjunction with rich, contextually-based prosodic features, to exploit sequential dependencies and to facilitate integration with lexical features. Integration of lexical and prosodic features improves pitch accent prediction over either feature set alone, and for lower accuracy feature sets, factorial CRF models can improve over linear chain based prediction of pitch accent.

1 Introduction

Prosody plays a crucial role in language understanding. In addition to the well-known effects in tone languages such as Chinese, prosody in English also plays a significant role, where pitch accents can indicate given/new information status, and boundary tones can distinguish statements from yes-no questions. However, recognition of such prosodic features poses significant challenges due to differences in surface realization from the underlying form. In particular, context plays a significant role in prosodic realization. Contextual effects due articulatory constraints such maximum speed of pitch change (Xu and Sun, 2002) from neighboring syllables and accents can yield co-articulatory effects at the intonational level, analogous to those at the segmental level. Recent phonetic research (Xu, 1999; Sun, 2002; Shen, 1990) has demonstrated the importance of coarticulation for tone and pitch accent recognition. In addition context affects interpretation of prosodic events; an accent is viewed as high or low relative to the speaker's pitch range and also relative to adjacent speech.

Some recent acoustically focused approaches (Sun, 2002; Levow, 2005) to tone and pitch accent recognition have begun to model and exploit these contextual effects on production. Following the Parallel Encoding and Target Approximation (PENTA) (Xu, 2004), this work assumes that the prosodic target is exponentially approached during the course of syllable production, and thus the target is best approximated in the later portion of the syllable. Other contextual evidence such as relative pitch height or band energy between syllables has also been employed (Levow, 2005; Rosenberg and Hirschberg, 2006). Interestingly, although earlier techniques (Ross and Ostendorf, 1994; Dusterhoff et al., 1999) employed Hidden Markov Models, they did not explicitly model these coarticulatory effects, and recent approaches have primarily employed local classifiers, such as decision trees (Sun, 2002; Rosenberg and Hirschberg, 2006) or Support Vector Machines (Levow, 2005).

Another body of work on pitch accent recognition has focused on exploitation of lexical and syntactic information to predict ToBI labels, for example for speech synthesis. These approaches explored a range of machine learning techniques from local classifiers such as decision trees (Sun, 2002) and RIPPER (Pan and McKeown, 1998) to sequence models such as Conditional Random Fields

(CRFs)(Gregory and Altun, 2004) more recently. The systems often included features that captured local or longer range context, such as n-gram probabilities, neighboring words, or even indicators of prior mention. (Chen et al., 2004; Rangarajan Sridhar et al., 2007) explored the integration of based prosodic and lexico-syntactic evidence in GMM-based and maximum entropy models respectively.

Here we explore the use of contextual acoustic and lexical models within a sequence learning framework. We analyze the interaction of different feature types on prediction of prosodic labels using linear-chain CRFs. We demonstrate improved recognition by integration of textual and acoustic cues, well-supported by the sequence model. Finally we consider the joint prediction of multiple prosodic label types, finding improvement for joint modeling in the case of feature sets with lower initial performance.

We begin by describing the ToBI annotation task and our experimental data. We then discuss the choice of conditional random fields and the use of linear chain and factorial models. Section 4 describes the contextual acoustic model and text-based features. Section 5 presents the experimental structure and results. We conclude with a brief discussion of future work.

2 Data

We employ a subset of the Boston Radio News Corpus (Ostendorf et al., 1995), employing data from speakers f1a, f2b, m1b, and m2b, for experimental consistency with (Chen et al., 2004; Rangarajan Sridhar et al., 2007). The corpus includes pitch accent, phrase and boundary tone annotation in the ToBI framework (Silverman et al., 1992) aligned with manual transcription and manual and automatic syllabification of the materials. Each word was also manually part-of-speech tagged. The data comprises over forty thousand syllables, with speaker f2b accounting for just over half the data. Following earlier research (Ostendorf and Ross, 1997; Sun, 2002), we collapse the ToBI pitch accent labels to four classes: unaccented, high, low, and downstepped high for experimentation, removing distinctions related to bitonal accents. We also consider the binary case of distinguishing accented from unaccented syllables, (Gregory and Altun, 2004; Rosenberg and Hirschberg, 2006; Ananthakrishnan and Narayanan, 2006). For phrase accents and boundary tones, we consider only the binary distinction between phrase accent/no phrase accent and boundary tone/no boundary tone.

All experiments evaluate automatic prosodic labeling at the syllable level.

3 Modeling with Linear-Chain and Factorial CRFs

Most prior acoustically based approaches to prosodic labeling have used local classifiers. However, on phonological grounds, we expect that certain label sequences will be much more probable than others. For example, sequences of multiple high accents are relatively uncommon in contrast to the case of an unaccented syllable preceding an accented one. This characteristic argues for a model which encodes and exploits inter-label dependencies. Furthermore, under the ToBI labeling guidelines, the presence of a boundary tone dictates the co-occurrence of a phrase accent label. To capture these relations between labels of different types, we also consider factorial models.

Conditional Random Fields (Lafferty et al., 2001) are a class of graphical models which are undirected and conditionally trained. While they can represent long term dependencies, most applications have employed first-order linear chains for language and speech processing tasks including POS tagging, sentence boundary detection (Liu et al., 2005), and even text-oriented pitch accent prediction(Gregory and Altun, 2004). The models capture sequential label-label relations, but unlike HMMs, the conditionally trained model can more tractably support larger text-based feature sets. Factorial CRFs (Sutton, 2006; McCallum et al., 2003) augment the linear sequence model with additional cotemporal labels, so that multiple (factors) labels are predicted at each time step and dependencies between them can be modeled. Examples of linear-chain and factorial CRFs appear in Figure 1. In the linear chain example, the f_i items correspond to the features and the y_i to labels to be predicted, for example prosodic and text features and pitch accent labels respectively. The vertical lines correspond to the dependencies

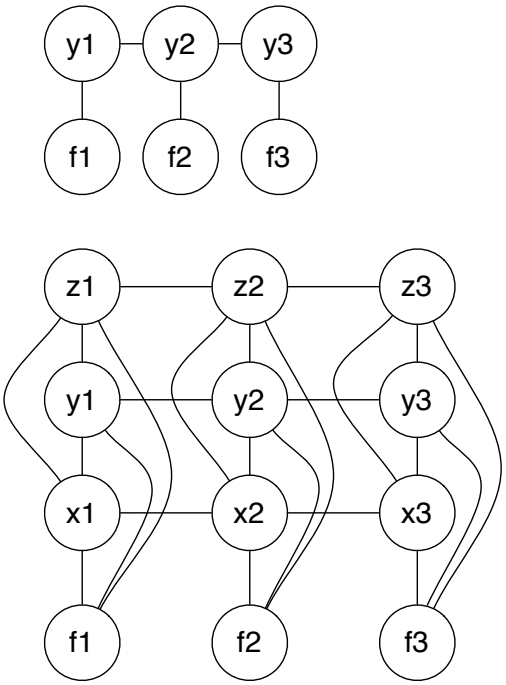


Figure 1: Linear-chain CRF (top) and Two-level Factorial CRF (bottom).

between the features and labels; the horizontal lines indicate the dependencies between the labels in sequence. In the factorial CRF example, the f_i again represent the features, while the x_i, y_i, and z_i represent the boundary tone, phrase accent, and pitch accent labels that are being predicted. The horizontal arcs again model the sequential bigram label-label dependencies between labels of the same class; the vertical arcs model the dependencies between both the features and labels, and bigram dependencies between the labels of each of the different pairs of factors. Thus, we jointly predict pitch accent, phrase accent, and boundary tone and, the prediction of each label depends on the features, the other labels predicted for the same syllable, and the sequential label of the same class. So, pitch accent prediction depends on the features, pitch accent predicted for the neighboring syllable, and phrase and boundary tone predictions for the current syllable.

We employ the Graphical Models for Mallet (GRMM) implementation (Sutton, 2006), adapted to also support the real-valued acoustic features required for these experiments; in some additional contrastive experiments on zero order models, we also employ the Mallet implementation (McCallum, 2002). We employ both linear chain and three-level factorial CRFs, as above, to perform prosodic labeling.

4 Feature Representation

We exploit both lexical and prosodic features for prosodic labeling of broadcast news speech. In particular, in contrast to (Gregory and Altun, 2004), we employ a rich acoustic feature set, designed to capture and compensate for coarticulatory influences on accent realization, in addition to word-based features.

4.1 Prosodic Features

Using Praat's (Boersma, 2001) "To pitch" and "To intensity" functions and the phoneme, syllable, and word alignments provided in the corpus, we extract acoustic features for the region of interest. This region corresponds to the syllable nucleus in English. For all pitch and intensity features, we compute per-speaker z-score normalized log-scaled values.

Recent phonetic research (Xu, 1997; Shih and Kochanski, 2000) has identified significant effects of carryover coarticulation from preceding adjacent syllable tones. To minimize these effects consistent with the pitch target approximation model (Xu et al., 1999), we compute slope features based on the second half of this region, where this model predicts that the underlying pitch height and slope targets of the syllable will be most accurately approached.

For each syllable, we compute the following local features:

- pitch values at five points evenly spaced across the syllable nucleus,
- mean and maximum pitch values,
- slope based on a linear fit to the pitch contour in the second half of the region, and
- mean and maximum intensity.

We consider two types of contextualized features as well, to model and compensate for coarticulatory effects from neighboring syllables. The first set of features, referred to as "extended features", includes the maximum and mean pitch from adjacent

syllables as well as the nearest pitch points from the adjacent syllables. These features extend the modeled tone beyond the strict bounds of the syllable segmentation. A second set of contextual features, termed "difference features", captures the change in feature values between the current and adjacent syllables. The resulting feature set includes:

- mean, maximum, and last two pitch values from preceding syllable,
- mean, maximum, and first value from following syllable, and
- differences in pitch mean, pitch maximum, pitch of midpoint, pitch slope, intensity mean, and intensity maximum between the current syllable and the preceding syllable, and between the current syllable and the following syllable.

Finally, we also employ some positional and durational features. Many prosodic phenomena are affected by phrase or sentence position; for example, both pitch and intensity tend to decrease across an utterance, and pitch accent realization may also be affected by cooccurring phrase accents or boundary tones. As syllable duration typically increases under both accenting and phrase-final lengthening, this information can be useful in prosodic labeling. Finally, pause information is also associated with prosodic phrasing. Thus, we include following features:

- two binary features indicating initial and final in a pseudo-phrase, defined as a silence-delimited interval,
- duration of syllable nucleus, and
- durations of pause preceding and following the syllable.

In prior experiments using support vector machines (Levow, 2005), variants of this representation achieved competitive recognition levels for both tone and pitch accent recognition.

4.2 Text-based Features

We employ text-based models similar to those employed by (Sun, 2002; Rangarajan Sridhar et al., 2007). For each syllable, we capture the following manually annotated features:

- The phonetic form of the current syllable, the previous two syllables, and the following two syllables,
- binary values indicating whether each of the current, previous, and following syllables are lexically stressed,
- integer values indicating position in a word of the current, previous, and following syllables,
- the current word, the two previous words, and the two following words, and
- the POS of the current word, of the two previous words, and of the two following words.

These features capture information about the current syllable and its lexico-syntactic context, that have been employed effectively in prosodic labeling of pitch accent, phrase accent, and boundary tone.

5 Experiments

We explore a range of issues in the experiments reported below. We hope to assess the impact of feature set and acoustic and text-based feature integration in the Conditional Random Field models. We compare their individual effectiveness as well as the effect of combined feature sets on labeling. In particular, we consider both the binary accented/unaccented assignment task for pitch accent and the four way - high/downstepped high/low/unaccented - contrast to compare effectiveness in problems of different difficulty. We further consider the effect of sequence and factorial modeling on pitch accent recognition. All experiments are conducted using a leave-one-out evaluation procedure following (Chen et al., 2004), training on all but one speaker and then testing on that held-out speaker, reporting the average across the tests on held-out data. Because speaker f2b contributes such a large portion of the data, that speaker is never left out.

On this split, the best word-based accuracy incorporating both prosodic and lexico-syntactic information in a maximum entropy framework is 86.0% for binary pitch accent prediction and 93.1% for

recognition of boundary status (Rangarajan Sridhar et al., 2007). For syllable-level recognition on this dataset, results for speaker-independent models reach slightly over 80% for binary pitch accent detection and 88% for boundary detection. Speaker dependent models have achieved very high accuracy; over 87% on speaker f2b was reported by (Sun, 2002) for the four-class task.

5.1 Explicit Prosodic Context Features and Sequence Models

We first assess the role of contextual prosodic features for pitch accent recognition and their interaction with sequence models. To minimize interaction effects, we concentrate on recognition with prosodic features alone on the challenging four-way pitch accent problem. As described above, we augmented the local syllable-based prosodic features with contextual features associated with the preceding and following syllables. We ask whether the use of contextual features improves recognition, and, if so, which type of context, preceding or following, has the greatest impact. We also ask whether the CRF models provide further improvements or can partially or fully compensate for the lack of explicit context features. To evaluate this impact, we compute four-way pitch accent recognition accuracy with no context features, after adding preceding context, after adding following context, and with both. We also contrast zero order and first order linear chain CRFs for these conditions. We find that modeling preceding context yields the greatest improvement. This finding is consistent with findings in recent phonetic research that argue for a larger role of carryover coarticulation from preceding syllables than of anticipatory coarticulation with following syllables. Furthermore, sequence modeling in the CRF also improves results, across the explicit context feature conditions, with improvements being most pronounced in cases with less effective explicit prosodic contextual features. Results for prosodic features alone appear in Table 1. In a side experiment with these prosodic features, we also briefly explored higher-order models, but no improvement was observed.

We also assess the impact of this richer contextualized prosodic feature set both alone and in conjunction with the full text-based feature set, in the full factorial CRF framework. We compare results for pitch accent identification in both the two-way and four-way conditions with no context and with the full ensemble of prosodic features. We find no difference for the two-way, all features condition for which text-based features perform well alone. However, for the prosody only cases and the more challenging four-way task with all features, contextual information yields improvements, demonstrating the utility of this richer, contextualized prosodic feature representation. These contrasts appear in Table 2.

		No Context	Full Context
Prosody	Two-way	78.9%	80.8%
Only	Four-way	74.2%	78.2%
All	Two-way	86.2%	86.2%
Features	Four-way	79%	79.7%

Table 2: Impact of context prosodic features with prosody alone and all features

5.2 Prosodic and Text-based Features

We continue by contrasting effectiveness of different feature sets in the basic linear-chain CRF case for pitch accent recognition. Table 3 presents the results for prosodic, word-based, and combined features sets in both the two-way and four-way classification conditions. Overall accuracy is quite good; in all cases, results are well above the 65% most common class assignment level, and the best results (86.2%) outperform any previously published speaker independent syllable-based results on this dataset. Overall results and contrasts are found in Table 3.

It is clear that the two feature sets combine very effectively. In the 4-way pitch accent task, the combined model yields a significant 1.5% to 2.5% increase over the strong acoustic-only model. In contrast, in the binary task, both the overall effectiveness of the text-based model and its utility in combination with the acoustic features are enhanced, yielding a much higher individual and combined accuracy rate. This contrast can be explained by the fact that the word features, such as part of speech, identify items that, as a class, are likely to be accented rather then being strongly associated with a particular tone category. The type of accent is likely

	No Context	Preceding	Following	Both
Zero order	70.5%	75.2%	71.8%	76.4%
First order	74.2%	75.5%	73.7%	77.1%

Table 1: Prosodic Context Features and CRFs

		Acoustic	Text	Text&Acoustic
Linear-Chain	Two-way	79.48%	84.88%	86.1%
	Four-way	77.06%	76.21%	79.65%
Factorial CRF	Two-way	80.76%	84.74%	86.2%
	Four-way	78.22%	77.46%	79.71%

Table 3: Pitch Accent Classification with Linear-Chain (top) and factorial CRFs (bottom) , using Acoustic-only, Text-based-only, and Combined Features. Results for two- and four-way pitch accent prediction are shown.

best determined by acoustic contrast, since accent type is closely linked to pitch height, and the local context and acoustic features serve to identify which accentable words are truly accented. Thus, in the binary task, the text-based features combine most effectively with the evidence from the acoustic features.

To contrast local classifiers with the linear chain model with text-based features, we trained a zero order classifier for the pitch accent prediction case and contrasted it with a comparable first-order linear-chain CRFs. Here for the binary accent recognition case, using only text-based information, we reach an accuracy of 84.3% for the history-free model, contrasted with an 85.4% level obtained with a comparable first-order model.[1]

5.3 Factorial CRF Framework

Finally we consider the effect of joint classification using the factorial CRF framework. Here, beyond just pitch accent assignment, we perform simultaneous assignment of pitch accent, phrase accent and boundary tone, where each label type corresponds to a factor, implementing the desired dependencies.[2]

[1]This comparison was computed using the original Mallet CRF package rather than GRMM, due to simpler zero order model support. This results in a small difference in the resulting scores.

[2]The features have not been tuned specifically for phrase account and boundary prediction, as explicit punctuation or sentence boundary features would have been useful but obvious giveaways. However, our goal is to assess the potential impact of combined classification, without excessive tuning.

The contrasts with the linear-chain model in terms of pitch accent prediction accuracy appear in Table 3. For the binary pitch accent condition, results are somewhat mixed. While there is a small but not significant decrease in accuracy for the text-only binary classification condition, the combined case shows little change and the prosodic case increases modestly. We note in one case that joint accuracy has risen when the pitch accent accuracy has dropped; we speculate that some additional compensation is needed to manage the effects of the severe class imbalance between the dominant "no-label" classes for phrase accent and boundary tone and other labels. For the four-way contrast between pitch accent types, we see small to modest gains across all feature sets, with the prosodic case improving significantly ($p < 0.025$). The best results for all but the two-way text-based classification task are found with the factorial CRF model.

For phrase accent and boundary tone prediction, phrase accent accuracy reaches 91.14%, and boundary tone accuracy 93.72% for all features. Text-based evidence is more effective than prosodic evidence in these cases, with text-based features reaching 91.06% for phrase accent and 92.51% and acoustic features only 86.73% and 92.37% respectively. However, little change is observed with the factorial CRF relative to a linear chain model trained on the same instances. The results for phrase accent and boundary tone recognition appear in Table 4.

	Phrase Accent	Boundary Tone
Prosodic	86.73%	92.37%
Text	91.06%	92.51%
Text+Prosodic	91.14%	93.72%

Table 4: Accuracy for phrase accent and boundary tone with prosodic, text-based, and combined features

6 Conclusion and Future Work

The application of linear-chain and factorial Conditional Random Fields for automatic pitch accent recognition and other prosodic labeling facilitates modeling of sequential dependencies as well as integration of rich acoustic features with text-based evidence. We plan to further investigate the modeling of dependencies between prosodic labels and the sequential modeling for acoustic features. Finally, we will also integrate prior work on subsyllable segmentation to identify the best approximation of the prosodic target with the CRF framework to produce a fine-grained sequence model of prosodic realization in context.

7 Acknowledgments

The author would like to thank Charles Sutton for providing the GRMM implementation, Andrew McCallum for the Mallet CRF implementation, and Siwei Wang and Sonja Waxmonsky for the modifications supporting real-valued features.

References

Sankaranarayanan Ananthakrishnan and Shrikanth Narayanan. 2006. Combining acoustic, lexical, and syntactic evidence for automatic unsupervised prosody labeling. In *Proceedings of ICSLP 2006*.

P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9–10):341–345.

K. Chen, M. Hasegawa-Johnson, and A. Cohen. 2004. An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model. In *Proceedings of ICASSP*.

K. Dusterhoff, A. Black, and P. Taylor. 1999. Using decision trees within the tilt intonation model to predict f0 contours. In *Proc. Of Eurospeech '99*.

Michelle Gregory and Yasemin Altun. 2004. Using conditional random fields to predict pitch accents in conversational speech. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 677–683, Barcelona, Spain, July.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML-2001)*.

Gina-Anne Levow. 2005. Context in multi-lingual tone and pitch accent prediction. In *Proc. of Interspeech 2005*.

Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. 2005. Using conditional random fields for sentence boundary detection in speech. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 451–458, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Andrew McCallum, Khashayar Rohanimanesh, and Charles Sutton. 2003. Dynamic conditional random fields for jointly labeling multiple sequences. In *NIPS*2003 Workshop on Syntax, Semantics, Statistics*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

M. Ostendorf and K. Ross. 1997. A multi-level model for recognition of intonation labels. In Y. Sagisaka, N. Campbell, and N. Higuchi, editors, *Computing Prosody*, pages 291–308.

M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel. 1995. The Boston University radio news corpus. Technical Report ECS-95-001, Boston University.

Shimei Pan and Kathleen McKeown. 1998. Learning intonation rules for concept to speech generation. In *Proceedings of ACL/COLING-98*, pages 1003–1009.

Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Shrikanth Narayanan. 2007. Exploiting acoustic and syntactic features for prosody labeling in a maximum entropy framework. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 1–8, Rochester, New York, April. Association for Computational Linguistics.

Andrew Rosenberg and Julia Hirschberg. 2006. On the correlation between energy and pitch accent in read english speech. In *Proceedings of ICLSP 2006*.

K. Ross and M. Ostendorf. 1994. A dynamical system model for generating f0 for synthesis. In *Proceedings of the ESCA/IEEE Workshop on Speech Synthesis*, pages 131–134.

Xiao-Nan Shen. 1990. Tonal co-articulation in Mandarin. *Journal of Phonetics*, 18:281–295.

C. Shih and G. P. Kochanski. 2000. Chinese tone modeling with stem-ml. In *Proceedings of the International Conference on Spoken Language Processing, Volume 2*, pages 67–70.

K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg. 1992. ToBI: A standard for labelling English prosody. In *Proceedings of ICSLP*, pages 867–870.

Xuejing Sun. 2002. Pitch accent prediction using ensemble machine learning. In *Proceedings of ICSLP-2002*.

Charles Sutton. 2006. Grmm: A graphical models toolkit. http://mallet.cs.umass.edu.

Yi Xu and X. Sun. 2002. Maximum speed of pitch change and how it may relate to speech. *Journal of the Acoustical Society of America*, 111.

C.X. Xu, Y. Xu, and L.-S. Luo. 1999. A pitch target approximation model for f0 contours in Mandarin. In *Proceedings of the 14th International Congress of Phonetic Sciences*, pages 2359–2362.

Yi Xu. 1997. Contextual tonal variations in Mandarin. *Journal of Phonetics*, 25:62–83.

Y. Xu. 1999. Effects of tone and focus on the formation and alignment of f0 contours - evidence from Mandarin. *Journal of Phonetics*, 27.

Yi Xu. 2004. Transmitting tone and intonation simultaneously - the parallel encoding and target approximation (PENTA) model. In *TAL-2004*, pages 215–220.

Chinese Unknown Word Translation by Subword Re-segmentation

Ruiqiang Zhang[1,2] and **Eiichiro Sumita**[1,2]
[1]National Institute of Information and Communications Technology
[2]ATR Spoken Language Communication Research Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan
{ruiqiang.zhang, eiichiro.sumita}@{nict.go.jp, atr.jp}

Abstract

We propose a general approach for translating Chinese unknown words (UNK) for SMT. This approach takes advantage of the properties of Chinese word composition rules, i.e., all Chinese words are formed by sequential characters. According to the proposed approach, the unknown word is re-split into a subword sequence followed by subword translation with a subword-based translation model. "Subword" is a unit between character and long word. We found the proposed approach significantly improved translation quality on the test data of NIST MT04 and MT05. We also found that the translation quality was further improved if we applied named entity translation to translate parts of unknown words before using the subword-based translation.

1 Introduction

The use of phrase-based translation has led to great progress in statistical machine translation (SMT). Basically, the mechanism of this approach is realized by two steps:training and decoding. In the training phase, bilingual parallel sentences are preprocessed and aligned using alignment algorithms or tools such as GIZA++ (Och and Ney, 2003). Phrase pairs are then extracted to be a phrase translation table. Probabilities of a few pre-defined features are computed and assigned to the phrase pairs. The final outcome of the training is a translation table consisting of source phrases, target phrases, and lists of probabilities of features. In the decoding phase, the translation of a test source sentence is made by reordering the target phrases corresponding to the source phrases, and searching for the best hypothesis that yields the highest scores defined by the search criterion.

However, this mechanism cannot solve unknown word translation problems. Unknown words (UNK) point to those unseen words in the training or non-existing words in the translation table. One strategy to deal with translating unknown words is to remove them from the target sentence without translation on assumption of fewer UNKs in the test data. Of course, this simple way produces a lower quality of translations if there are a lot of UNKs in the test data, especially for using a Chinese word segmenter that produces many UNKs. The translation of UNKs need to be solved by a special method.

The translation of Chinese unknown words seems more difficult than other languages because Chinese language is a non-inflected language. Unlike other languages (Yang and Kirchhoff, 2006; Nießlen and Ney, 2000; Goldwater and McClosky, 2005), Chinese UNK translation cannot use information from stem and inflection analysis. Using machine transliteration can resolve part of UNK translation (Knight and Graehl, 1997). But this approach is effective for translating phonetically related unknown words, not for other types. No unified approach for translating Chinese unknown words has been proposed.

In this paper we propose a novel statistics-based approach for unknown word translation. This approach uses the properties of Chinese word composition rules – Chinese words are composed of one or more Chinese characters. We can split longer unknown words into a sequence of smaller units: characters or subwords. We train a subword based translation model and use the model to translate the sub-

word sequence. Thus we get the translation of the UNKs. We call this approach "subword-based unknown word translation".

In what follows, section 2 reviews phrase-based SMT. section 3 describes the dictionary-based CWS, that is the main CWS in this work. Section 4 describes our named entity recognition approach. Section 5 describes the subword-based approach for UNK translation. Section 7 describes the experiments we conducted to evaluate our subword approach for translating Chinese unknown words. Section 8 describes existing methods for UNK translations for other languages than Chinese. Section 9 briefly summarizes the main points of this work.

2 Phrase-based statistical machine translation

Phrase-based SMT uses a framework of log-linear models (Och, 2003) to integrate multiple features. For Chinese to English translation, source sentence C is translated into target sentence E using a probability model:

$$P_\Lambda(E|C) = \frac{exp(\sum_{i=1}^{M} \lambda_i f_i(C,E))}{\sum_{E'} exp(\sum_{i=1}^{M} \lambda_i f_i(C,E'))} \quad \Lambda = \{\lambda_1^M,\} \tag{1}$$

where $f_i(C,E)$ is the logarithmic value of the i-th feature, and λ_i is the weight of the i-th feature. The candidate target sentence that maximizes $P(E|C)$ is the solution.

Obviously, the performance of such a model depends on the qualities of its features. We used the following features in this work.

- Target language model: an N-gram language model is used.
- Phrase translation model $p(e|f)$: gives the probability of the target phrases for each source phrase.
- Phrase inverse probability $p(f|e)$: the probability of a source phrase for a given target phrase. It is the coupled feature of the last one.
- Lexical probability $lex(e|f,a)$: the sum of the target word probabilities for the given source words and the alignment of the phrase pairs.
- Lexical inverse probability $lex(f|e,a)$: the sum of the source word probabilities for the given target words and alignment.
- Target phrase length model $\#(p)$: the number of phrases included in the translation hypothesis.
- Target word penalty model: the number of words included in the translation hypothesis.
- Distance model $\#(w)$: the number of words between the tail word of one source phrase and the head word of the next source phrase.

In general, the following steps are used to get the above features.

1. Data processing: segment Chinese words and tokenize the English.
2. Word alignment: apply two-way word alignment using GIZA++.
3. Lexical translation: calculate word lexical probabilities.
4. Phrase extraction: extract source target bilingual pairs by means of union, intersection, et. al.
5. Phrase probability calculation: calculate phrase translation probability.
6. Lexical probability: generate word lexical probabilities for phrase pairs.
7. Minimal error rate training: find a solution to the λ's in the log-linear models.

3 Dictionary-based Chinese word segmentation

For a given Chinese character sequence, $C = c_0c_1c_2\ldots c_N$, the problem of word segmentation is addressed as finding a word sequence, $W = w_{t_0}w_{t_1}w_{t_2}\ldots w_{t_M}$, where the words, $w_{t_0}, w_{t_1}, w_{t_2}, \ldots, w_{t_M}$, are pre-defined by a provided lexicon/dictionary, which satisfy

$$\begin{aligned} &w_{t_0} = c_0\ldots c_{t_0}, \quad w_{t_1} = c_{t_0+1}\ldots c_{t_1} \\ &w_{t_i} = c_{t_{i-1}+1}\ldots c_{t_i}, \quad w_{t_M} = c_{t_{M-1}+1}\ldots c_{t_M} \\ &t_i > t_{i-1}, \quad 0 \le t_i \le N, \quad 0 \le i \le M \end{aligned}$$

This word sequence is found by maximizing the function below,

$$\begin{aligned} W &= \arg\max_W P(W|C) \\ &= \arg\max_W P(w_{t_0} w_{t_1} \dots w_{t_M}) \end{aligned} \quad (2)$$

We applied Bayes' law in the above derivation. $P(w_{t_0} w_{t_1} \dots w_{t_M})$ is a language model that can be expanded by the chain rule. If trigram LMs are used, it is approximated as

$$P(w_0)P(w_1|w_0)P(w_2|w_0w_1)\cdots P(w_M|w_{M-2}w_{M-1})$$

where w_i is a shorthand for w_{t_i} .

Equation 2 indicates the process of the dictionary-based word segmentation. Our CWS is based on it. We used a beam search algorithm because we found that it can speed up the decoding. Trigram LMs were used to score all the hypotheses, of which the one with the highest LM scores is the final output.

As the name indicates, the word segmentation results by the dictionary-based CWS are dependent on the size and contents of the lexicon. We will use three lexicons in order to compare effects of lexicon size to the translations. The three lexicons denoted as Character, Subword and Hyperword are listed below. An example sentence, 黄英春住在北京市(HuangYingChun lives in Beijing City), is given to show the segmentation results of using the lexicons.

- Character: Only Chinese single characters are included in the lexicon. The sentence is split character by character. 黄/英/春/住/在/北/京/市
- Subword: A small amount of most frequent words (10,000) are added to the lexicon. Choosing the subwords are described in section 5. 黄/英/春/住/在/北京/市
- Hyperword: A big size of lexicon is used, consisting of 100,000 words. 黄/英/春/住/在/北京市

4 Named entity recognition (NER)

Named entities in the test data need to be treated separately. Otherwise, a poor translation quality was found by our experiments. We define four types of named entities: people names (nr), organization names (nt), location names (ns), and numerical expressions (nc) such as calendar, time, and money. Our NER model is built according to conditional random fields (CRF) methods (Lafferty et al., 2001), by which we convert the problem of NER into that of sequence labeling. For example, we can label the last section's example as, "黄/B_nr 英/I_nr 春/I_nr 住/O 在/O 北/B_nt 京/I_nt 市/I_nt", where "B" stands for the first character of a NE; "I", other than the first character of a NE; "O", isolated character. "nr" and "nt" are two labels of NE.

Table 1: NER accuracy

type	Recall	Precision	F-score
nr	85.32%	93.41%	89.18%
ns	87.80%	90.46%	89.11%
nt	84.50%	87.54%	85.99%
all	84.58%	90.97%	87.66%

We use the CRF++ tools to train the models for named entity recognition[1]. The performance of our NER model was shown in Table 4. We use the Peking University (PKU) named entity corpus to train the models. Part of the data was used as test data.

We stick to the results of CWS if there are ambiguities in the segmentation boundary between CWS and NER.

The NER was used only on the test data in translations. It was not used on the training data due to the consideration of data sparseness. Using NER will generate more unknown words that cannot be found a translation in the translation table. That is why we use a subword-based translation approach.

5 Subword-based translation model for UNK translation

We found there were two reasons accounting for producing untranslatable words. The first is the size of lexicon. We proposed three size of lexicons in section 3, of which the Hyperword type uses 100,000 words. Because of a huge lexical size, some of the words cannot be learned by SMT training because of limited training data. The CWS chooses only one candidate segmentation from thousands in

[1]http://chasen.org/~taku/software/CRF++/

splitting a sentence into word sequences. Therefore, the use of a candidate will block other candidates. Hence, many words in the lexicon cannot be fully trained if a large lexicon is used. The second is our NER module. The NER groups a longer sequence of characters into one entity that cannot be translated. We have analyzed this points in the last section.

Therefore, in order to translate unknown words, our approach is to split longer unknown words into smaller pieces, and then translate the smaller pieces by using Character or Subword models. Finally, we put the translations back to the Hyperword models. We call this method subword-based unknown word translation regardless of whether a Character model or Subword model is used.

As described in Section 3, Characters CWS uses only characters in the lexicon. So there is no tricks for it. But for the Subword CWS, its lexicon is a small subset of the Hyperword CWS. In fact, we use the following steps for generating the lexicon. In the beginning, we use the Hyperword CWS to segment the training data. Then, we extract a list of unique tokens and calculate their counts from the results of segmentation. Next, we sort the list as the decreasing order of the counts, and choose N most frequent words from the top of the list. We restrict the length of subwords to three. We use the N words as the lexicon for the subword CWS. N can be changed. Section 7.4 shows its effect to translations. The subword CWS uses a trigram language model to disambiguate. Refer to (Zhang et al., 2006) for details about selecting the subwords.

We applied Subword CWS to re-segment the training data. Finally, we can train a subword-based SMT translation model used for translating the unknown words. Training this subword translation model was done in the same way as for the Hyperword translation model that uses the main CWS, as described in the beginning of Section 2.

6 Named entity translation

The subword-based UNK translation approach can be applied to all the UNKs indiscriminately. However, if we know an UNK is a named entity, we can translate this UNK more accurately than using the subword-based approach. Some unknown words can be translated by named entity translation if they are correctly recognized as named entity and fit a translation pattern. For example, the same words with different named entities are translated differently in the context. The word, "九", is translated into "nine" for measures and money, "September" for calendar, and "jiu" for Chinese names.

As stated in Section 4, we use NER to recognize four types of named entities. Correspondingly, we created the translation patterns to translate each type of the named entities. These patterns include patterns for translating numerical expressions, patterns for translating Chinese and Japanese names, and patterns for translating English alphabet words. The usages are described as follows.

Numerical expressions are the largest proportion of unknown words. They include calendar-related terms (days, months, years), money terms, measures, telephone numbers, times, and addresses. These words are translated using a rule-based approach. For example, "三点十五分", is translated into "at 3:15".

Chinese and Japanese names are composed of two, three, or four characters. They are translated into English by simply replacing each character with its spelling. The Japanese name, "安倍晋三", is translated into "Shinzo Abe".

English alphabets are encoded in different Chinese characters. They are translated by replacing the Chinese characters with the corresponding English letters.

We use the above translation patterns to translate the named entities. Using translation patterns produce almost correct translation. Hence, we put the named entity translation to work before we apply the subword translation model. The subword translation model is used when the unknown words cannot be translated by named entity translation.

7 SMT experiments

7.1 Data

We used LDC Chinese/English data for training. We used two test data of NIST MT04 and NIST MT05. The statistics of the data are shown in Table 6. We used about 2.4 million parallel sentences extracted from LDC data for training. Experiments on both the MT04 and MT05 test data used the same translation models on the same training data, but the min-

Table 2: Statistics of data for MT experiments

			Chinese	English
MT	Training	Sentences	2,399,753	
		words	49,546,231	52,746,558
MT04 LDC2006E43	Test	Sentences	1,788	
		Words	49,860	
MT05 LDC2006E38	Test	Sentences	1,082	
		Words	30,816	

Table 3: Statistics of unknown words of test data using different CWS

	Hyperword+Named entities					Hyperword	Subwords	Characters
	Numerics	People	Org.	Loc.	other			
MT04	460	146	250	230	219	650	18	2
MT05	414	271	311	146	323	680	23	2

imum error rate training was different. The MT04 and MT05 test data were also used as development data for cross experiments.

We used a Chinese word segmentation tool, Achilles, for doing word segmentation. Its word segmentation accuracy was higher than the stanford word segmenter (Tseng et al., 2005) in our laboratory test (Zhang et al., 2006).

The average length of a sentence for the test data MT04 and MT05 after word segmentation is 37.5 by using the Subword CWS, and 27.9 by using the Hyperword CWS.

Table 6 shows statistics of unknown words in MT04 and MT05 using different word segmentation. Obviously, character-based and subword-based CWS generated much fewer unknown words, but sentences are over-segmented. The CWS of Hyperword generated many UNKs because of using a large size of lexicon. However, if named entity recognition was applied upon the segmented results of the Hyperword, more UNKs were produced. Take an example for MT04. There are 1,305 UNKs in which numeric expressions amount to 35.2%, people names at 11.2%, organization names at 19.2%, location names at 17.6%, and others at 16.8%. Analysis of these numbers helps to understand the distribution of unknown words.

7.2 Effect of the various CWS

As described in section 3, we used three lexicon size for the dictionary-based CWS. Therefore, we had three CWS denoted as: Character, Subword and Hyperword. We used the three CWS in turn to do word segmentation to the training data, and then built the translation models respectively. We tested the performance of each of the translation models on the test data. The results are shown on Table 4. The translations are evaluated in terms of BLEU score (Papineni et al., 2002). This experiment was just testing the effect of the three CWS. Therefore, all the UNKs of the test data were not translated, simply removed from the results.

We found the character-based CWS yielded the lowest BLEU scores, indicating the translation quality of this type is the worst. The Hyperword CWS achieved the best results. If we relate it to Table 6, we found while the Hyperword CWS produced many more UNKs than the Character and Subword CWS, its translation quality was improved instead. The fact proves the quality of translation models play a more important role than the amount of unknown word translation. Using the Hyperword CWS can generate a higher quality of translation models than the Character and Subword CWS. Therefore, we cannot use the character and subword-based CWS in Chinese SMT system due to their overall poor performance. But we found their

Table 4: Compare the translations by different CWS (BLEU scores)

	MT04	MT05
Character	0.253	0.215
Subword	0.265	0.229
Hyperword	0.280	0.236

Table 5: Effect of subword and named entity translation (BLEU)

	MT04	MT05
Baseline(Hyperword)	0.280	0.236
Baseline+Subword	0.283	0.244
Baseline+NER	0.283	0.242
Baseline+NER+Subword	0.285	0.246

usage for UNK translation.

7.3 Effect of subword translation for UNKs

The experiments in this section show the effect of using the subword translation model for UNKs. We compared the results of using subword translation with those of without using it. We also used named entity translation together with the subword translation. Thus, we could compare the effect of subword translation under conditions of with or without named entity translation. We listed four kinds of results to evaluate the performance of our approach in Table 5 where the symbols indicate:

- *Baseline:* this is the results made by the Hyperword CWS of Table 4. No subword translation for UNKs and named entity translations were used. Unknown words were simply removed from the output.

- *Baseline+Subword:* the results were made under the same conditions as the first except all of the UNKs were extracted, re-segmented by the subword CWS and translated by the subword translation models. However, the named entity translation was not used.

- *Baseline+NER:* this experiment did not use subword-based translation for UNKs. But we used named entity translation. Part of UNKs was labeled with named entities and translated by pattern match of section 6.

- *Baseline+NER+Subword:* this experiment used the named entity translation and the subword-based translation. The difference from the second one is that some UNKs were translated by the translation patterns of section 6 at first and the remaining UNKs were translated using the subword model (the second one translated all of the UNKs using the subword model).

The results of our experiments are shown in Table 5. We found the subword models improved translations in all of the experiments. Using the subword models on the MT04 test data improved translations in terms of BLEU scores from 0.280 to 0.283, and from 0.236 to 0.244 on the MT05 test data. While only small gains of BLEU were achieved by UNK translation, this improvement is sufficient to prove the effectiveness of the subword models, given that the test data had only a low proportion of UNKs.

The BLEU scores of "Baseline+NER" is higher than that of "Baseline", that proves using named entity translation improved translations, but the effect of using named entity translation was worse than using the subword-based translation. This is because the named entity translation is applicable for the named entities only. However, the subword-based translation is used for all the UNKs.

When we applied named entity translation to translate some of recognized named entities followed by using the subword models, we found BLEU gains over using the subword models uniquely, 0.2% for MT04 and 0.2% for MT05. This experiment proves that the best way of using the subword models is to separate the UNKs that can be translated by named entity translation from those that cannot, and let the subword models handle translations of those not translated.

Analysis using the bootstrap tool created by Zhang et al. (Zhang et al., 2004) showed that the results made by the subword translations were significantly better than the ones not using it.

7.4 Effect of changing the size of subword lexicon

We have found a significant improvement by using the subword models. The essence of the approach

Table 6: BLEU scores for changing the subword lexicon size

subword size	MT04	MT05
character	0.280	0.237
10K	0.283	0.244
20K	0.283	0.240

is to split unknown words into subword sequences and use subword models to translate the subword sequences. The choices are flexible in choosing the number of subwords in the subword lexicon. If a different subword list is used, the results of the subword re-segmentation will be changed. Will choosing a different subword list have a large impact on the translation of UNKs? As shown in Table 6, we used three classes of subword lists: character, 10K subwords and 20K subwords. The "character" class used only single-character words, about 5,000 characters. The other two classes, "10K" and "20K", used 10,000 and 20,000 subwords. The method for choosing the subwords was described in Section 5. We have used "10K" in the previous experiments. We did not use named entity translation for this experiment.

We found that using "character" as the subword unit brought in nearly no improvement over the baseline results. Using 20K subwords yielded better results than the baseline but smaller gains than that of using the 10K subwords for MT05 data. It proves that using subword translation is an effective approach but choosing a right size of subword lexicon is important. We cannot propose a better method for finding the size. We can do more experiments repeatedly to find this value. We found the size of 10,000 subwords achieved the best results for our experiments.

8 Related work

Unknown word translation is an important problem for SMT. As we showed in the experiments, appropriate handling of this problem results in a significant improvement of translation quality. As we have known, there exists some methods for solving this problem. While these approaches were not proposed in aim to unknown word translation, they can be used for UNK translations indirectly.

Most existing work focuses on named entity translation (Carpuat et al., 2006) because named entities are the large proportion of unknown words. We also used similar methods for translating named entities in this work.

Some used stem and morphological analysis for UNKs such as (Goldwater and McClosky, 2005). Morphological analysis is effective for inflective languages but not for Chinese. Using unknown word modeling such as backoff models was proposed by (Yang and Kirchhoff, 2006).

Other proposed methods include paraphrasing (Callison-Burch et al., 2006) and transliteration (Knight and Graehl, 1997) that uses the feature of phonetic similarity. However, This approach does not work if no phonetic relationship is found.

Splitting compound words into translatable subwords as we did in this work have been used by (Nießlen and Ney, 2000) and (Koehn and Knight, 2003) for languages other than Chinese where detailed splitting methods are proposed. We used forward maximum match method to split unknown words. This splitting method is relatively simple but works well for Chinese. The splitting for Chinese is not as complicated as those languages with alphabet.

9 Discussion and conclusion

We made use of the specific property of Chinese language and proposed a subword re-segmentation to solve the translation of unknown words. Our approach was tested under various conditions such as using named entity translation and varied subword lexicons. We found this approach was very effective. We are hopeful that this approach can be applied into languages that have similar features as Chinese, for example, Japanese.

While the work was done on a SMT system which is not the state-of-the-art [2], the idea of using subword-based translation for UNKs is applicable to any systems because the problem of UNK translation has to be faced by any system.

Acknowledgement

The authors would like to thank Dr.Michael Paul for his assistance in this work, especially for evaluating methods and statistical significance test.

[2]The BLEU score of the top one system is about 0.35 for MT05 (http://www.nist.gov/speech/tests/mt/).

References

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *HLT-NAACL-2006*.

Marine Carpuat, Yihai Shen, Xiaofeng Yu, and Dekai Wu. 2006. Toward Integrating Word Sense and Entity Disambiguation into Statistical Machine Translation. In *Proc. of the IWSLT*.

Sharon Goldwater and David McClosky. 2005. Improving statistical MT through morphological analysis. In *Proceedings of the HLT/EMNLP*.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. In *Proc. of the ACL*.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *EACL-2003*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML-2001*, pages 591–598.

Sonja Nießlen and Hermann Ney. 2000. Improving smt quality with morpho-syntactic analysis. In *Proc. of COLING*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

F. J. Och. 2003. Minimum error rate training for statistical machine trainslation. In *Proc. ACL*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the 40th ACL*, pages 311–318, Philadelphia, USA.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for Sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, Jeju, Korea.

Mei Yang and Katrin Kirchhoff. 2006. Phrase-based backoff models for machine translation of highly inflected languages. In *EACL-2006*.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system? In *Proceedings of the LREC*.

Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 2006. Subword-based tagging by conditional random fields for chinese word segmentation. In *Proceedings of the HLT-NAACL*.

Hypothesis Selection in Machine Transliteration: A Web Mining Approach

Jong-Hoon Oh and Hitoshi Isahara
Computational Linguistics Group
National Institute of Information and Communications Technology (NICT)
3-5 Hikaridai,Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{rovellia,isahara}@nict.go.jp

Abstract

We propose a new method of selecting hypotheses for machine transliteration. We generate a set of Chinese, Japanese, and Korean transliteration hypotheses for a given English word. We then use the set of transliteration hypotheses as a guide to finding relevant Web pages and mining contextual information for the transliteration hypotheses from the Web page. Finally, we use the mined information for machine-learning algorithms including support vector machines and maximum entropy model designed to select the correct transliteration hypothesis. In our experiments, our proposed method based on Web mining consistently outperformed systems based on simple Web counts used in previous work, regardless of the language.

1 Introduction

Machine transliteration has been a great challenge for cross-lingual information retrieval and machine translation systems. Many researchers have developed machine transliteration systems that accept a source language term as input and then output its transliteration in a target language (Al-Onaizan and Knight, 2002; Goto et al., 2003; Grefenstette et al., 2004; Kang and Kim, 2000; Li et al., 2004; Meng et al., 2001; Oh and Choi, 2002; Oh et al., 2006; Qu and Grefenstette, 2004). Some of these have used the Web to select machine-generated transliteration hypotheses and have obtained promising results (Al-Onaizan and Knight, 2002; Grefenstette et al., 2004; Oh et al., 2006; Qu and Grefenstette, 2004). More precisely, they used simple Web counts, estimated as the number of hits (Web pages) retrieved by a Web search engine.

However, there are several limitations imposed on the ability of Web counts to select a correct transliteration hypothesis. First, the assumption that hit counts approximate the Web frequency of a given query usually introduces noise (Lapata and Keller, 2005). Moreover, some Web search engines disregard punctuation and capitalization when matching search terms (Lapata and Keller, 2005). This can cause errors if such Web counts are relied on to select transliteration hypotheses. Second, it is not easy to consider the contexts of transliteration hypotheses with Web counts because Web counts are estimated based on the number of retrieved Web pages. However, as our preliminary work showed (Oh et al., 2006), transliteration or translation pairs often appear as parenthetical expressions or tend to be in close proximity in texts; thus context can play an important role in selecting transliteration hypotheses. For example, there are several Chinese, Japanese, and Korean (CJK) transliterations and their counterparts in a parenthetical expression, as follows.

1) 阿德里安娜$_1$克拉克森$_2$ (Adrienne$_1$ Clarkson$_2$)

2) グルコース$_1$オキシダーゼ$_2$ (glucose$_1$ oxidase$_2$)

3) 디페놀$_1$ 옥시다아제$_2$ (diphenol$_1$ oxidase$_2$)

Note that the subscripted numbers in all examples represent the correspondence between the English word and its CJK counterpart. These parenthetical expressions are very useful in selecting translit-

eration hypotheses because it is apparent that they are translation pairs or transliteration pairs. However, we cannot fully use such information with Web counts.

To address these problems, we propose a new method of selecting transliteration hypotheses. We were interested in how to mine information relevant to the selection of hypotheses and how to select correct transliteration hypotheses using the mined information. To do this, we generated a set of CJK transliteration hypotheses for a given English word. We then used the set of transliteration hypotheses as a guide to finding relevant Web page and mining contextual information for the transliteration hypotheses from the Web page. Finally, we used the mined information for machine-learning algorithms including support vector machines (SVMs) and maximum entropy model designed to select the correct transliteration hypothesis.

This paper is organized as follows. Section 2 describes previous work based on simple Web counts. Section 3 describes a way of generating transliteration hypotheses. Sections 4 and 5 introduce our methods of Web mining and selecting transliteration hypotheses. Sections 6 and 7 deal with our experiments and the discussion. Conclusions are drawn and future work is discussed in Section 8.

2 Related work

Web counts have been used for selecting transliteration hypotheses in several previous work (Al-Onaizan and Knight, 2002; Grefenstette et al., 2004; Oh et al., 2006; Qu and Grefenstette, 2004). Because the Web counts are estimated as the number of hits by a Web search engine, they greatly depend on queries sent to a search engine. Previous work has used three types of queries—*monolingual queries* (MQs) (Al-Onaizan and Knight, 2002; Grefenstette et al., 2004; Oh et al., 2006), *bilingual simple queries* (BSQs) (Oh et al., 2006; Qu and Grefenstette, 2004), and *bilingual bigram queries* (BBQs) (Oh et al., 2006). If we let S be a source language term and $\mathcal{H} = \{h_1, \cdots, h_r\}$ be a set of machine-generated transliteration hypotheses of S, the three types of queries can be defined as

MQ: h_i (e.g., 克林頓 , クリントン, and 클린턴).

BSQ: s and h_i without quotations (e.g., *Clinton* 克林頓 , *Clinton* クリントン, and *Clinton* 클린턴).

BBQ: Quoted bigrams composed of S and h_i (e.g., "*Clinton* 克林頓", "*Clinton* クリントン", and "*Clinton* 클린턴").

MQ is not able to determine whether h_i is a counterpart of S, but whether h_i is a frequently used target term in target-language texts. BSQ retrieves Web pages if S and h_i are present in the same document but it does not take the distance between S and h_i into consideration. BBQ retrieves Web pages where "S h_i" or "h_i S" are present as a bigram. The relative order of Web counts over $\mathcal{H}$ makes it possible to select transliteration hypotheses in the previous work.

3 Generating Transliteration Hypotheses

Let S be an English word, P be a pronunciation of S, and T be a target language transliteration corresponding to S. We implement English-to-CJK transliteration systems based on three different transliteration models — a grapheme-based model ($S \rightarrow T$), a phoneme-based model ($S \rightarrow P$ and $P \rightarrow T$), and a correspondence-based model ($S \rightarrow P$ and $(S, P) \rightarrow T$) — as described in our preliminary work (Oh et al., 2006). P and T are segmented into a series of sub-strings, each of which corresponds to a source grapheme. We can thus write $S = s_1, \cdots, s_n = s_1^n$, $P = p_1, \cdots, p_n = p_1^n$, and $T = t_1, \cdots, t_n = t_1^n$, where s_i, p_i, and t_i represent the i^{th} English grapheme, English phonemes corresponding to s_i, and target language graphemes corresponding to s_i, respectively. Given S, our transliteration systems generate a sequence of t_i corresponding to either s_i (in Eq. (1)) or p_i (in Eq. (2)) or both of them (in Eq. (3)).

$$Pr_G(T|S) = Pr(t_1^n|s_1^n) \quad (1)$$

$$Pr_P(T|S) = Pr(p_1^n|s_1^n) \times Pr(t_1^n|p_1^n) \quad (2)$$

$$Pr_C(T|S) = Pr(p_1^n|s_1^n) \times Pr(t_1^n|s_1^n, p_1^n) \quad (3)$$

The maximum entropy model was used to estimate probabilities in Eqs. (1)–(3) (Oh et al., 2006). We produced the n-best transliteration hypotheses using a stack decoder (Schwartz and Chow, 1990). We

then created a set of transliteration hypotheses comprising the n-best transliteration hypotheses.

4 Web Mining

Let S be an English word and $\mathcal{H} = \{h_1, \cdots, h_r\}$ be its machine-generated set of transliteration hypotheses. We use S and $\mathcal{H}$ to generate queries sent to a search engine[1] to retrieve the top-100 snippets. A correct transliteration and its counterpart tend to be in close proximity on CJK Web pages. Our goal in Web mining was to find such Web pages and mine information that would help to select transliteration hypotheses from these pages.

To find these Web pages, we used three kinds of queries, Q_1=(S and h_i), Q_2=S, and Q_3=h_i, where Q_1 is the same as BSQ's query and Q_3 is the same as MQ's. The three queries usually result in different sets of Web pages. We categorize the retrieved Web pages by Q_1, Q_2, and Q_3 into W_1, W_2, and W_3. We extract three kinds of features from W_l as follows, where $l = 1, 2, 3$.

- $Freq(h_i, W_l)$: the number of occurrences of h_i in W_l
- $DFreq_k(h_i, W_l)$: Co-occurrence of S and h_i with distance $d_k \in D$ in the same snippet of W_l.
- $PFreq_k(h_i, W_l)$: Co-occurrence of S and h_i as parenthetical expressions with distance $d_k \in D$ in the same snippet of W_l. Parenthetical expressions are detected when either S or h_i is in parentheses.

We define $D = \{d_1, d_2, d_3\}$ with three ranges of distances between S and h_i, where $d_1(d < 5)$, $d_2(5 \leq d < 10)$, and $d_3(10 \leq d \leq 15)$. We counted distance d with the total number of characters (or words)[2] between S and h_i. Here, we can take the contexts of transliteration hypotheses into account using $DFreq$ and $PFreq$; while $Freq$ is counted regardless of the contexts of the transliteration hypotheses.

Figure 1 shows examples of how to calculate $Freq$, $DFreq_k$, and $PFreq_k$, where $S = Clinton$, h_i=克林頓 in W_1 collected by Q_1=(*Clinton* 克林頓). The subscripted numbers of *Clinton* and 克林頓 were used to indicate how many times they occurred in W_1. In Fig. 1, 克林頓 occurs six times thus $Freq(h_i, W_1) = 6$. Table 1 lists the distance between *Clinton* and 克林頓 within each snippet of W_1. We can obtain $DFreq_1(h_i, W_1) = 5$. $PFreq_1(h_i, W_l)$ is calculated by detecting parenthetical expressions between S and h_i when $DFreq_1(h_i, W_l)$ is counted. Because all S in W_1 ($Clinton_1$ to $Clinton_5$) are in parentheses, $PFreq_1(h_i, W_1)$ is the same as $DFreq_1(h_i, W_1)$.

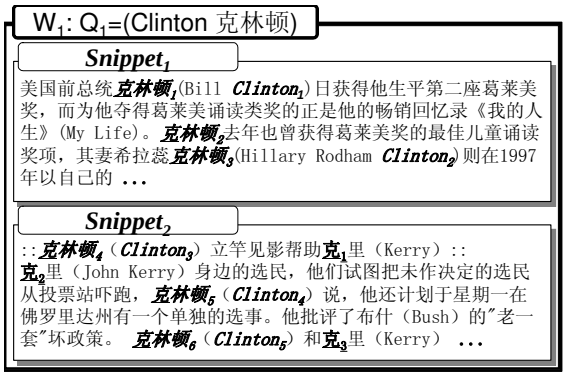


Figure 1: Web corpora collected by *Clinton* and 克林頓

$Snippet_1$	克林頓$_1$	克林頓$_2$	克林頓$_3$
$Clinton_1$	1	41	68
$Clinton_2$	72	29	2
$Snippet_2$	克林頓$_4$	克林頓$_5$	克林頓$_6$
$Clinton_3$	0	36	81
$Clinton_4$	40	0	37
$Clinton_5$	85	41	0
$Snippet_2$	克$_1$	克$_2$	克$_3$
$Clinton_3$	6	9	85
$Clinton_4$	32	29	42
$Clinton_5$	77	74	1

Table 1: Distance between $Clinton$ and Chinese transliteration hypotheses in Fig. 1

We ignore $Freq$, $DFreq_k$, and $PFreq_k$ when h_i is a substring of other transliteration hypotheses because h_i usually has a higher $Freq$, $DFreq_k$, and $PFreq_k$ than h_j if h_i is a substring of h_j. Let a

[1]We used Google (`http://www.google.com`)

[2]Depending on whether the languages had spacing units, words (for English and Korean) or characters (for Chinese and Japanese) were chosen to calculate d.

set of transliteration hypotheses for $S = Clinton$ be $\mathcal{H}$= {h_1 = 克林頓, h_2 = 克}. Here, h_2 is a substring of h_1. In Fig. 1, h_2 appears six times as a substring of h_1 and three times independently in $Snippet_2$. Moreover, independently used h_2 (克$_1$, 克$_2$, and 克$_3$) and S ($Clinton_3$ and $Clinton_5$) are sufficiently close to count $DFreq_k$ and $PFreq_k$. Therefore, the $Freq$, $DFreq_k$, and $PFreq_k$ of h_1 will be lower than those of h_2 if we do not take the substring relation between h_1 and h_2 into account. Considering the substring relation, we obtain $Freq(h_2, W_1) = 3$, $DFreq_1(h_2, W_1) = 1$, $DFreq_2(h_2, W_1) = 2$, $PFreq_1(h_2, W_1) = 1$, and $PFreq_2(h_2, W_1) = 2$.

5 Hypothesis Selection

We select transliteration hypotheses by ranking them. A set of transliteration hypotheses, $\mathcal{H} = \{h_1, h_2, \cdots, h_r\}$, is ranked to enable a correct hypothesis to be identified. We devise a rank function, $g(h_i)$ in Eq. (4), that ranks a correct transliteration hypothesis higher and the others lower.

$$g(h_i) : \mathcal{H} \rightarrow \{\mathcal{R} : \mathcal{R}\ is\ ordering\ of\ h_i \in \mathcal{H}\} \quad (4)$$

Let $x_i \in \mathcal{X}$ be a feature vector of $h_i \in \mathcal{H}$, $y_i = \{+1, -1\}$ be the training label for x_i, and $\mathcal{TD} = \{td_1 =< x_1, y_1 >, \cdots, td_z =< x_z, y_z >\}$ be the training data for $g(h_i)$. We prepare the training data for $g(h_i)$ as follows.

1. Given each English word S in the *training-set*, generate transliteration hypotheses $\mathcal{H}$.

2. Given $h_i \in \mathcal{H}$, assign y_i by looking for S and h_i in the *training-set* — $y_i = +1$ if h_i is a correct transliteration hypothesis corresponding to S, otherwise $y_i = -1$.

3. For each pair (S, h_i), generate its feature vector x_i.

4. Construct a training data set, $\mathcal{TD}$:

   - $\mathcal{TD} = \mathcal{TD}^+ \bigcup \mathcal{TD}^-$
   - $\mathcal{TD}^+ \ni td_i$ where $y_i = +1$
   - $\mathcal{TD}^- \ni td_j$ where $y_j = -1$

We used two machine-learning algorithms, support vector machines (SVMs)[3] and maximum entropy model[4] for our implementation of $g(h_i)$. The SVMs assign a value to each transliteration hypothesis (h_i) using

$$g_{SVM}(h_i) = w \cdot x_i + b \quad (5)$$

where w denotes a weight vector. Here, we use the predicted value of $g_{SVM}(h_i)$ rather than the predicted class of h_i given by SVMs because our ranking function, as represented by Eq. (4), determines the relative ordering between h_i and h_j in $\mathcal{H}$. A ranking function based on the maximum entropy model assigns a probability to h_i using

$$g_{MEM}(h_i) = Pr(y_i = +1|x_i) \quad (6)$$

We can finally obtain a ranked list for the given $\mathcal{H}$— the higher the $g(h_i)$ value, the better the h_i.

5.1 Features

We represent the feature vector, x_i, with two types of features. The first is the confidence scores of h_i given by Eqs. (1)–(3) and the second is Web-based features — $Freq$, $DFreq_k$, and $PFreq_k$. To normalize $Freq$, $DFreq_k$, and $PFreq_k$, we use their relative frequency over $\mathcal{H}$ as in Eqs. (7)–(9), where $k = 1, 2, 3$ and $l = 1, 2, 3$.

$$RF(h_i, W_l) = \frac{Freq(h_i, W_l)}{\sum_{h_j \in \mathcal{H}} Freq(h_j, W_l)} \quad (7)$$

$$RDF_k(h_i, W_l) = \frac{DFreq_k(h_i, W_l)}{\sum_{h_j \in \mathcal{H}} DFreq_k(h_j, W_l)} \quad (8)$$

$$RPF_k(h_i, W_l) = \frac{PFreq_k(h_i, W_l)}{\sum_{h_j \in \mathcal{H}} PFreq_k(h_j, W_l)} \quad (9)$$

Figure 2 shows how to construct feature vector x_i from a given English word, *Rachel*, and its Chinese hypotheses, $\mathcal{H}$, generated from our transliteration systems. We can obtain r Chinese transliteration hypotheses and classify them into positive and negative samples according to y_i. Note that $y_i = +1$ if and only if h_i is registered as a counterpart of S in the training data. The bottom of Fig. 2 shows our feature set representing x_i. There are three confidence scores in $P(h_i|S)$ according to transliteration models and the three Web-based features $Web(W_1)$, $Web(W_2)$, and $Web(W_3)$.

[3] SVM^{light} (Joachims, 2002)

[4] "Maximum Entropy Modeling Toolkit" (Zhang, 2004)

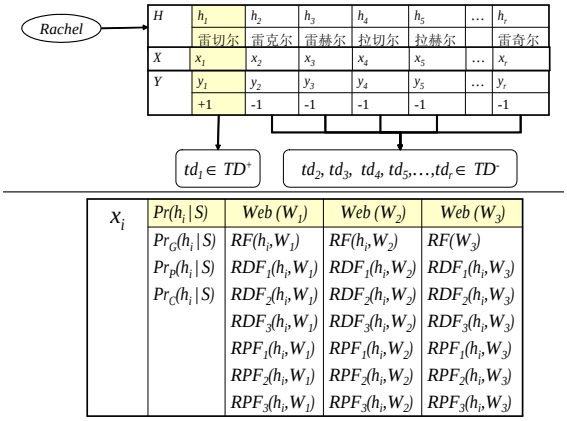


Figure 2: Feature vectors

6 Experiments

We evaluated the effectiveness of our system in selecting CJK transliteration hypotheses. We used the same test set used in Li et al. (2004) (ECSet) for Chinese transliterations (Xinhua News Agency, 1992) and those used in Oh et al. (2006) for Japanese and Korean transliterations — EJSET and EKSET (Breen, 2003; Nam, 1997). We divided the test data into training, development, and blind test sets as in Table 2. The training set was used to train our three transliteration models to generate the n-best transliteration hypotheses[5]. The development set was used to train hypothesis selection based on support vector machines and maximum entropy model.

	ECSet	EJSet	EKSet
Training Set	31,299	8,335	5,124
Development Set	3,478	1,041	1,024
Blind Test Set	2,896	1,041	1,024
Total	37,694	10,417	7,172

Table 2: Test data sets

We used the blind test set for evaluation. The evaluation was done in terms of word accuracy (WA). WA is the proportion of correct transliterations in the best hypothesis by a system to correct transliterations in the blind test set.

System	ECSet	EJSet	EKSet
KANG00	N/A	N/A	54.1
GOTO03	N/A	54.3	N/A
LI04	70.1	N/A	N/A
GM	69.0	61.6	59.0
PM	56.6	54.4	56.7
CM	69.9	65.0	65.1

Table 3: WA of individual transliteration systems (%)

6.1 Results: Web counts vs. Web mining

We compared our transliteration system with three previous ones, all of which were based on a grapheme-based model (Goto et al., 2003; Kang and Kim, 2000; Li et al., 2004). LI04[6] is an English-to-Chinese transliteration system, which simultaneously takes English and Chinese contexts into consideration (Li et al., 2004). KANG00 is an English-to-Korean transliteration system and GOTO03 is an English-to-Japanese one – they segment a chunk of English graphemes and identify the most relevant sequence of target graphemes corresponding to the chunk (Goto et al., 2003; Kang and Kim, 2000) [7]. GM, PM, and CM, which are respectively based on Eqs. (1)–(3), are the transliteration systems we used for generating transliteration hypotheses. Our transliteration systems showed comparable or better performance than the previous ones regardless of the language.

We compared simple Web counts with our Web mining for hypothesis selection. We used the same set of transliteration hypotheses $\mathcal{H}$ then compared their performance in hypothesis selection with two measures, relative frequency and $g(h_i)$. Tables 4 and 5 list the results. Here, "Upper bound" is a system that always selects the correct transliteration hypothesis if there is a correct one in $\mathcal{H}$. "Upper bound" can

[5]We set $n = 10$ for the n-best. Thus, $n \leq r \leq 3 \times n$ where $\mathcal{H} = \{h_1, h_2, \cdots, h_r\}$

[6]The WA of LI04 was taken from the literature, where the training data were the same as the union of our training set and the development set while the test data were the same as in our test set. In other words, LI04 used more training data than ours did. With the same setting as LI04, our GM, PM, and CM produced respective WAs of 70.0, 57.7, and 71.7.

[7]We implemented KANG00 (Kang and Kim, 2000) and GOTO03 (Goto et al., 2003), and tested them with the same data as ours.

System		ECSet	EJSet	EKSet
WC	MQ	16.1	40.4	34.7
	BSQ	45.8	74.0	72.4
	BBQ	34.9	78.1	79.3
WM	$RF(W_1)$	62.9	78.4	77.1
	$RDF(W_1)$	70.8	80.4	80.2
	$RPF(W_1)$	73.5	79.7	79.4
	$RF(W_2)$	63.5	76.2	74.8
	$RDF(W_2)$	67.1	79.2	78.9
	$RPF(W_2)$	69.6	79.1	78.4
	$RF(W_3)$	37.9	53.9	55.8
	$RDF(W_3)$	76.4	69.0	70.2
	$RPF(W_3)$	76.8	68.3	68.7
Upper bound		94.6	93.5	93.2

Table 4: Web counts (WC) vs. Web mining (WM): hypothesis selection by relative frequency (%)

System		ECSet	EJSet	EKSet
WC	MEM_{WC}	74.7	86.1	85.6
	SVM_{WC}	74.8	86.9	86.5
WM	MEM_{WM}	82.0	88.2	85.8
	SVM_{WM}	83.9	88.5	86.7
Upper bound		94.6	93.5	93.2

Table 5: Web counts (WC) vs. Web mining (WM): hypothesis selection by $g(h_i)$ (%)

also be regarded as the "Coverage" of $\mathcal{H}$ generated by our transliteration systems. MQ, BSQ, and BBQ in the upper section of Table 4, represent hypothesis selection systems based on the relative frequency of Web counts over $\mathcal{H}$, the same measure used in Oh et al. (2006):

$$\frac{WebCounts_x(h_i)}{\sum_{h_j \in \mathcal{H}} WebCounts_x(h_j)} \quad (10)$$

where $WebCounts_x(h_i)$ is a function returning Web counts retrieved by $x \in \{MQ, BSQ, BBQ\}$ $RF(W_l)$, $RDF(W_l)$, and $RPF(W_l)$ in Table 4 represent hypothesis selection systems with their relative frequency, where $RDF(W_l)$ and $RPF(W_l)$ use $\sum_{k=1}^{3} RDF_k(h_j, W_l)$ and $\sum_{k=1}^{3} RPF_k(h_j, W_l)$, respectively. The comparison in Table 4 shows which is best for selecting transliteration hypotheses when each relative frequency is used alone. Table 5 compares Web counts with features mined from the Web when they are used as features in $g(h_i)$ — $\{Pr(h_i|S), Web(W_l)\}$ in MEM_{WM} and SVM_{WM} (our proposed method), while $\{Pr(h_i|S), WebCounts_x(h_i)\}$ in MEM_{WC} and SVM_{WC}. Here, $Web(W_l)$ is a set of mined features from W_l as described in Fig .2.

Figure 3: Snippets causing errors in Web counts

The results in the tables show that our systems consistently outperformed systems based on Web counts, especially for Chinese. This was due to the difference between languages. Japanese and Chinese do not use spaces between words. However, Japanese is written using three different alphabet systems, called *Hiragana*, *Katakana*, and *Kanji*, that assist word segmentation. Moreover, words written in *Katakana* are usually Japanese transliterations of foreign words. This makes it possible for a Web search engine to effectively retrieve Web pages containing given Japanese transliterations. Like English, Korean has spaces between words (or word phrases). As the spaces in the languages reduce ambiguity in segmenting words, a Web search engine can correctly identify Web pages containing given Korean transliterations. In contrast, there is a severe word-segmentation problem with Chinese that causes Chinese Web search engines to incorrectly retrieve Web pages, as shown in Fig. 3. For example, $Snippet_1$ is not related to "Aman" but to "a man".

$Snippet_2$ contains a super-string of a given Chinese query, which corresponds to "Academy" rather than to "Agard", which is the English counterpart of the Chinese transliteration 阿加. Moreover, Web search engines ignore punctuation marks in Chinese. In $Snippet_3$ and $Snippet_4$, "," and "·" in the underlined terms are disregarded, so the Web counts based on such Web documents are noisy. Thus, noise in the Chinese Web counts causes systems based on Web counts to produce more errors than our systems do. Our proposed method can filter out such noise because our systems take punctuation marks and the contexts of transliterations in Web mining into consideration. Thus, our systems based on features mined from the Web were able to achieve the best performance. The results revealed that our systems based on the Web-mining technique can effectively be used to select transliteration hypotheses regardless of the language.

6.2 Contribution of Web corpora

	ECSet		EJSet		EKSet	
	SVM	MEM	SVM	MEM	SVM	MEM
Base	73.3	73.8	67.0	66.1	66.0	66.4
W_1	81.7	79.7	87.6	87.3	86.1	85.1
W_2	80.8	79.5	86.9	86.0	83.8	82.1
W_3	77.2	76.7	83.0	82.8	79.8	77.3
W_{1+2}	83.8	82.3	88.5	87.9	86.3	85.9
W_{1+3}	81.9	80.1	87.6	87.8	86.1	84.7
W_{2+3}	81.4	79.8	88.0	87.7	85.1	84.3
W_{All}	83.9	82.0	88.5	88.2	86.7	85.8

Table 6: Contribution of Web corpora

In Web mining, we used W_1, W_2, and W_3, collected by respective queries Q_1=(S and h_i), Q_2=S, and Q_3=h_i. To investigate their contribution, we tested our proposed method with different combinations of Web corpora. "Base" is a baseline system that only uses $Pr(h_i|S)$ as features but does not use features mined from the Web. We added features mined from different combinations of Web corpora to "Base" from W_1 to W_{All}.

In Table 6, we can see that W_1, a set of Web pages retrieved by Q_1, tends to give more relevant information than W_2 and W_3, because Q_1 can search more Web pages containing both S and h_i in the top-100 snippets if S and h_i are a correct transliteration pair. Therefore, its performance tends to be superior in Table 6 if W_1 is used, especially for ECSet. However, as W_1 occasionally retrieves few snippets, it is not able to provide sufficient information. Using W_2 or W_3, we can address the problem. Thus, combinations of W_1 and others (W_{1+2}, W_{1+3}, W_{All}) provided better WA than W_1.

7 Discussion

Several Web mining techniques for transliteration lexicons have been developed in the last few years (Jiang et al., 2007; Oh and Isahara, 2006). The main difference between ours and those previous ones is in the way a set of transliteration hypotheses (or candidates) is created.

Jiang et al. (2007) generated Chinese transliterations for given English words and searched the Web using the transliterations. They generated only the best transliteration hypothesis and focused on Web mining to select transliteration lexicons rather than selecting transliteration hypotheses. The best transliteration hypothesis was used to guide Web searches. Then, transliteration candidates were mined from the retrieved Web pages. Therefore, their performance greatly depended on their ability to mine transliteration candidates from the Web. However, this system might create errors if it cannot find a correct transliteration candidate from the retrieved Web pages. Because of this, their system's coverage and WA were relatively poor than ours [8]. However, our transliteration process was able to generate a set of transliteration hypotheses with excellent coverage and could thus achieve superior WA.

Oh and Isahara (2006) searched the Web using given source words and mined the retrieved Web pages to find target-language transliteration candidates. They extracted all possible sequences of target-language characters from the retrieved Web snippets as transliteration candidates for which the beginnings and endings of the given source word

[8] Since both Jiang et al.'s (2007) and ours used Chinese transliterations of personal names as a test set, we can indirectly compare our coverage and WA with theirs (Jiang et al., 2007). Jiang et al. (2007) achieved a 74.5% coverage of transliteration candidates and 47.5% WA, while ours achieved a 94.6% coverage of transliteration hypotheses and 82.0–83.9% WA

and the extracted transliteration candidate were phonetically similar. However, while this can exponentially increase the number of transliteration candidates, ours used the n-best transliteration hypotheses but still achieved excellent coverage.

8 Conclusion

We have described a novel approach to selecting transliteration hypotheses based on Web mining. We first generated CJK transliteration hypotheses for a given English word and retrieved Web pages using the transliteration hypotheses and the given English word as queries for a Web search engine. We then mined features from the retrieved Web pages and trained machine-learning algorithms using the mined features. Finally, we selected transliteration hypotheses by ranking them. Our experiments revealed that our proposed method worked well regardless of the language, while simple Web counts were not effective, especially for Chinese.

Because our method was very effective in selecting transliteration pairs, we expect that it will also be useful for selecting translation pairs. We plan to extend our method in future work to selecting translation pairs.

References

Y. Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In *Proc. of ACL '02*, pages 400–408.

J. Breen. 2003. EDICT Japanese/English dictionary .le. The Electronic Dictionary Research and Development Group, Monash University. `http://www.csse.monash.edu.au/~jwb/edict.html`.

I. Goto, N. Kato, N. Uratani, and T. Ehara. 2003. Transliteration considering context information based on the maximum entropy method. In *Proc. of MT-Summit IX*, pages 125–132.

Gregory Grefenstette, Yan Qu, and David A. Evans. 2004. Mining the Web to create a language model for mapping between English names and phrases and Japanese. In *Proc. of Web Intelligence*, pages 110–116.

Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu. 2007. Named entity translation with Web mining and transliteration. In *Proc. of IJCAI*, pages 1629–1634.

Thorsten Joachims. 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers.

I. H. Kang and G. C. Kim. 2000. English-to-Korean transliteration using multiple unbounded overlapping phoneme chunks. In *Proc. of COLING '00*, pages 418–424.

Mirella Lapata and Frank Keller. 2005. Web-based models for natural language processing. *ACM Trans. Speech Lang. Process.*, 2(1):3.

H. Li, M. Zhang, and J. Su. 2004. A joint source-channel model for machine transliteration. In *Proc. of ACL '04*, pages 160–167.

H.M. Meng, Wai-Kit Lo, Berlin Chen, and K. Tang. 2001. Generating phonetic cognates to handle named entities in English-Chinese cross-language spoken document retrieval. In *Proc. of Automatic Speech Recognition and Understanding, 2001. ASRU '01*, pages 311–314.

Y. S. Nam. 1997. *Foreign dictionary*. Sung An Dang.

Jong-Hoon Oh and Key-Sun Choi. 2002. An English-Korean transliteration model using pronunciation and contextual rules. In *Proc. of COLING2002*, pages 758–764.

Jong-Hoon Oh and Hitoshi Isahara. 2006. Mining the Web for transliteration lexicons: Joint-validation approach. In *Web Intelligence*, pages 254–261.

Jong-Hoon Oh, Key-Sun Choi, and Hitoshi Isahara. 2006. A comparison of different machine transliteration models. *Journal of Artificial Intelligence Research (JAIR)*, 27:119–151.

Yan Qu and Gregory Grefenstette. 2004. Finding ideographic representations of Japanese names written in Latin script via language identification and corpus validation. In *Proc. of ACL '04*, pages 183–190.

Richard Schwartz and Yen-Lu Chow. 1990. The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypothesis. In *Procs. of ICASSP '90*, pages 81–84.

Xinhua News Agency. 1992. *Chinese transliteration of foreign personal names*. The Commercial Press.

L. Zhang. 2004. Maximum entropy modeling toolkit for python and C++. `http://homepages.inf.ed.ac.uk/s0450736/software/maxent/manual.pdf`.

What Prompts Translators to Modify Draft Translations? An Analysis of Basic Modification Patterns for Use in the Automatic Notification of Awkwardly Translated Text

Takeshi Abekawa and Kyo Kageura
Library and Information Science Course
Graduate School of Education,
University of Tokyo
{abekawa,kyo}@p.u-tokyo.ac.jp

Abstract

In human translation, translators first make draft translations and then modify them. This paper analyses these modifications, in order to identify the features that trigger modification. Our goal is to construct a system that notifies (English-to-Japanese) volunteer translators of awkward translations. After manually classifying the basic modification patterns, we analysed the factors that trigger a change in verb voice from passive to active using SVM. An experimental result shows good prospects for the automatic identification of candidates for modification.

1 Introduction

We are currently developing an English-to-Japanese translation aid system aimed at volunteer translators mainly working online (Abekawa and Kageura, 2007), As part of this project, we are developing a module that notifies (inexperienced) translators of awkwardly translated expressions that may need refinement or editing.

In most cases, translators first make draft translations, and then examine and edit them later, often repeatedly. Thus there are normally at least two versions of a given translation, i.e. a draft and the final translation. In commercial translation environments, it is sometimes the case that texts are first translated by inexperienced translators and then edited by experienced translators. However, this does not apply to voluntary translation. In addition, volunteer translators tend to be less experienced than commercial translators, and devote less time to editing. It would therefore be of great help to these translators if the CAT system automatically pointed out awkward translations for possible modification. In order to realise such a system, it is necessary to first clarify (i) the basic types of modification made by translators to draft translations, and (ii) what triggers these modifications.

In section 2 we introduce the data used in this study. In section 3, we clarify the nature of modification in the translation process. In section 4, we identify the actual modification patterns in the data. In section 5, focusing on "the change from the passive to the active voice" pattern, we analyse and clarify the triggers that may lead to modification. Section 6 is devoted to an experiment in which machine learning methods are used to detect modification candidates. The importance of the various triggers is examined, and the performance of the system is evaluated.

2 The data

The data used in the present study is the Japanese translation of an English book about the problem of peak oil (Leggett, 2005). The book is aimed at a popular audience and is relevant to the sort of texts we have in mind, because the majority of texts volunteer translators translate deal with current affairs, social issues, politics, culture and sports, and/or economic issues for a popular audience[1]. The data consists of the English original (henceforth "English"), the draft Japanese translation ("Draft") and the final Japanese translation ("Final"). The "Draft" was made by two translators (one with two years' experience and the other with five years' experience), and

[1] Software localisation is another area of translation in which volunteers are heavily involved. We do not include it in our target because it has different characteristics.

Figure 1: An example of word alignment using GIZA++

the "Final" was made by a translator with 12 years' experience. Table 1 gives the quantities of the data.

	"English"	"Draft"	"Final"
Number of sentences	4,587	4,629	4,648
Number of words	92,300	127,838	132,989
(Average per sentence)	20.1	27.6	28.6

Table 1: Basic quantities of the data

3 Nature of the modification process

	State	Cause
1.	Mistranslation	English is complex
2.	Text is confusing	English is complex / Translation is too literal
3.	Text is unnatural	Translation is too literal / Japanese is underexamined
4.	Against modifiers' taste	Different Japanese 'model' is assumed
5.	Against editorial policy	Lack of surface editing

Table 2: States in the draft and their causes

As little research has been carried out into the process by which translators modify draft translations, we manually analysed a part of the data in which modifications were made, in consultation with a translator. In the modification process, the translator first recognises (though often not consciously) one of a number of *states* in a draft translation and the underlying cause of the state. S/he then modifies the draft translation if necessary. Table 2 shows the basic classification of *states* and possible *causes*. Although the states are conceptually clear, it is not necessarily the case that translators can judge the state of a given translation consistently, because judging a sentence as being "natural" or "confusing" is not a binary process but a graded one, and the distinction between different states is often not immediately clear. Many concrete modification patterns found in the data are covered in translation textbooks (Anzai, 1995; Nakamura, 2003). However, although it is obvious in some cases that a section of translated text needs to be modified, in other cases it is less clear, and judgments will vary according to the translator. The task that automatic notification addresses, therefore, is essentially an ambiguous one, even though the actual system output may be binary.

We also identified the distinction between two types of modification: (i) "generative" modification, in which the modified translation is generated on the spot, with reference to the English original; and (ii) "considered" modification, in which alternate expressions (phrases, collocations, etc.) are retrieved from the depository of useful, elegant, or conventional expressions in the translator's mind. These two types of modification can be activated in the face of one token of modification at once.

4 Modification patterns

The most natural way to classify *modification patterns* is by means of basic linguistic labels such as "change of voice" or "change from nominal modification to adverbial modification" (cf. Anzai, 1995). These modification patterns consist of one or more *primitive operations*. For instance, a "change of voice" may consist of such primitive operations as "changing the case-marker of the subject," "swapping the position of subject and object," etc.

As preparation, we extracted modification patterns from the data[2]. In order to do so, we first aligned the "Draft" and the "Final" at the sentence level using DP matching, and then at the morpheme level using GIZA++ (Och and Ney, 2003). Figure 1 illustrates an example of word/morpheme level

[2]This task is similar to the acquisition of paraphrase knowledge (Barzilay and McKeown, 2001; Shinyama et al., 2002; Quirk et al. 2004; Barzilay and Lee, 2003; Dolan et al., 2004). However, our aim here is to clarify basic modification patterns and not automatic identification.

English:	If it was perceived to be true by the majority of Thinkers, ...				
"Draft":	人類の JINRUI-NO (thinkers$_{genitive}$)	多数によって TASUU-NIYOTTE (majority$_{ablative}$)	それが SORE-GA (it$_{subject}$)	真実であると SINJITU-DE-ARU-TO (to be true)	認識されれば, NINSIKI-SA-RERE-BA (be perceived)
"Final":	人類の JINRUI-NO (thinkers$_{genitive}$)	多数が TASUU-GA (majority$_{subject}$)	それを SORE-WO (it$_{object}$)	真実と SINJITU-TO (to be true)	認識すれば, NINSIKI-SURE-BA (perceive)
Primitive operations:		replace("NIYOTTE", "GA")	replace("GA", "WO")	delete("DE") delete("ARU")	delete("RARERU")

Table 3: An example of a primitive modification operation

alignment. Changes in word order occur frequently, as is shown in Figure 1, and the "Final" and the "Draft" are not completely parallel at the word or morpheme level. As a result, GIZA++ sometimes misaligns the units.

From the aligned "Draft" and "Final" data, we identified the primitive operations. We limited these operations to syntactic operations and semantic operations such as the changing of content words, because the latter is hard to generalise with a small amount of data. Primitive operations were extracted by calculating the difference between corresponding *bunsetsu*, which basically consist of a content word and postpositions/suffixes, in the "Draft" and in the "Final". An example is given in Table 3. Table 4 shows the five most frequent changes in verb inflections and case markers, which are two dominant classes of primitive operation. In addition, we observed deletions and insertions of Sahen verbs.

Modification patterns were identified by observing the degree of co-occurrence among these primitive operations. We used Cabocha[3] to identify the syntactic dependencies and used the log-likelihood ratio (LLR) to calculate the degree of co-occurrence of primitive operations that occupy syntactically dependent positions. Table 5 shows the top five pairwise co-occurrence patterns.

inflection	del.	ins.	case marker	del.	ins.
DA	379	291	NI	476	384
TE	269	358	GA	387	502
TA	247	306	NO	366	204
RARERU	224	122	WO	293	421
IRU	197	267	DE	203	193

Table 4: Frequent primitive operations

[3]http://chasen.org/~taku/software/cabocha/

Three main modification patterns were identified: (i) a change from the passive to the active voice (226 cases); (ii) a change from a Sahen verb to a Sahen noun (208 cases); and (iii) a change from nominal modification to clausal structure. These patterns have been discussed in studies of paraphrases (Inui and Fujita, 2004) and in translation textbooks (Anzai, 1995; Nakamura, 2003). We focus on "the change from the passive to the active voice". It is one of the most important and interesting modification patterns because (i) it is mostly concerned with the main clausal structure in which other modifications are embedded; and (ii) the use of active and passive voices differs greatly between English and Japanese and thus there will be much to reveal.

5 Triggers that lead to modification

Given a draft translation, an experienced translator will be able to recognise any problematic states in it (see Table 2), identify the causes of these states and deal with them. As computers (and inexperienced translators) cannot do the same (cf. Sun et al., 2007), it is necessary to break these causes down into computationally tractable triggers. Keeping in mind the nature of the modification process discussed in section 3, we analysed the actual data, this time with the help of a translator and a linguist.

At the topmost level, two types of triggers were identified: (i) "pushing" triggers that are identified as negative characteristics of the draft translation expressions themselves; and (ii) "pulling" triggers that come from outside (from the depository of expressions in the translator's mind) and work as concrete "model translations". The distinction is not entirely clear, because a model is needed in order to identify negative characteristics, and some sort of negative impression is needed for the "model translation" to be called up. The distinction is nevertheless

LLR	f(a,b)	f(a)	f(b)	operation a	operation b	plain expression
146.2	28	35	224	replace(NIYOTTE,GA)	delete(RARERU)	A NIYOTTE B SARERU→A GA B SURU
105.2	34	90	224	replace(GA,WO)	delete(RARERU)	A GA B SARERU→A WO B SURU
91.7	34	115	208	replace(NO,GA)	delete(SAHEN)	A NO B→A GA B SURU
90.9	26	61	208	replace(NO,WO)	delete(SAHEN)	A NO B→A WO B SURU
36.3	15	68	168	replace(NI,WO)	intransitive→transitive	A NI B SURU→A WO C SURU

Table 5: Five of the most frequent co-occurrence patterns between two primitive operations

important, both theoretically and practically. Theoretically, it corresponds to the types of modification observed in section 3. From the practical point of view, the first type is related to the general structural modelling (in its broad sense) of language, while the second is closely related to the status of individual lexicalised expressions. Correspondingly, an NLP system that addresses the first type needs to assume a language model, while a system that addresses the second type needs to call on the relevant external data on the spot. We address the first type of trigger, because we can hypothesise that the modification by change of voice is mainly related to the structural nature of expressions. It should also be noted that, from the machine learning point of view, there are positive and negative features which respectively promote and restrict the modification.

We classified the features that may represent potential triggers into five groups:

(A) Features related to the readability of the English, because the complexity of English sentences (cf. Fry, 1968; Gunning, 1959) can affect the quality of draft translations. Thus the number of words in a sentence, length of words, number of verbs in a sentence, number of commas, etc. can be used as tractable features for automatic treatment.

(B) Features reflecting the correspondence between the English and the draft Japanese translation. Translations that are very literal, either lexically or structurally, are often also awkward. On the other hand, a high degree of word order correspondence can be a positive sign (cf. Anzai, 1995), because it indicates that the information flow in English is maintained and the Japanese translation is well examined.

(C) Features related to the Japanese target verbs. The characteristics of the target verbs should affect the environments in which they occur.

(D) Features related to the "naturalness" of the Japanese. Repetitions or redundancies of elements or sound patterns may lead to unnatural Japanese sentences.

(E) Features related to the complexity of the Japanese. If a draft translation is too complex, it may be confusing or hard to read. Structural complexity, the length of a sentence, the number of commas, etc. can be used as triggers that reflect the complexity of the Japanese translation.

Table 6 shows the computationally tractable features we defined within this framework. Features with '#' in their name are numeric features and the others are binary features (taking either 0 or 1).

6 Detecting modification candidates

Using these features, we carried out an experiment of automatic identification of modification candidates. As a machine learning method, we used SVM (Vapnik, 1995). The aim of the experiment was twofold: (i) to observe the feasibility of automatic notification of modification candidates, and (ii) to examine the factors that trigger modifications in more detail.

6.1 Experimental setup

In the application of SVM, we reduced the number of binary features by using those that have higher correlations with positive and negative examples, using mutual information (MI). Table 7 shows features that have high correlations with positive and negative features (eight for each).

SVM settings: The liner kernel was used. For a numeric feature X, the value x is normalized by z-score, $norm(x) = \frac{x - avg(X)}{\sqrt{var(X)}}$, where $avg(x)$ is the empirical mean of X and $var(X)$ is the variance of X.

Data: The numbers of positive and negative cases in the data are 226 and 894, respectively (1120 in total). In order to balance the positive and negative examples, we used an equal number of examples for training.

(A)

$EN_{\#word}$:	the number of words in the English sentence
$EN_{\#pause}$:	the number of delimiters in the English sentence
$EN_{\#verb}$:	the number of verbs in the English sentence
$EN_{\#VVN}$:	the number of VNN verbs in the English sentence
$EN_{\#word\_len}$:	the average number of characters in a word

(B)

E_{POS}:	POS of the English word corresponding to the target Japanese verb
E_{POS_before}:	POS of a word before the English word corresponding to the target Japanese verb
E_{POS_after}:	POS of a word after the English word corresponding to the target Japanese verb
$E_{POS\_before:POS}$:	a bigram of E_{POS_before} and E_{POS}
$E_{POS:POS\_after}$:	a bigram of E_{POS} and E_{POS_after}
$EJ_{\#translation}$:	translation probability between the source and target language sentences

(C)

F_{suffix}:	a suffix following the target verb
$F_{particle}$:	a particle following the target verb
F_{pause_park}:	a pause mark following the target verb
$D_{modifying\_case}$:	case marker of the element that modifies the target verb
$D_{modifying\_agent}$:	case marker of the element that modifies the target verb, if its case element has an AGENT attribute
$D_{functional}$:	functional noun which is modified by the target verb
$D_{modified\_case}$:	case marker of the element that is modified by the target verb
S_{first_agent}:	first case element in the sentence has an AGENT attribute
$S_{before\_passive}$:	Is there a passive verb before the target verb in the sentence?
$S_{after\_passive}$:	Is there a passive verb after the target verb in the sentence?

(D)

$N_{modifying\_voice}$:	the voice of the verb that modifies the target verb
$N_{modifying\_voice}$:	the voice of the verb that is modified by the target verb
$N_{grandparent\_voice}$:	the voice of the grandparent verb of the target verb
$N_{grandchild\_voice}$:	the voice of the grandchild verb of the target verb
$N_{case\_adjacency}$;	bigram consists of a particle of the target verb and a particle of the adjacency bunsetsu chunk

(E)

$J_{\#morpheme}$:	the number of morphemes in the target Japanese sentence
$J_{\#pause}$:	the number of pause marks in the target Japanese sentence
$J_{\#verb}$:	the number of verbs in the target Japanese sentence
$J_{\#passive}$:	the number of verbs with passive voice in the target Japanese sentence
$J_{\#depth}$:	depth of the modifier which modifies the target verb

Table 6: Features

Methods of evaluation: We used (i) 10-fold cross validation to check the power of classifiers for unknown data and (ii) a partially closed test in which the 226 positive and negative examples were used for training and 1120 data were evaluated, in order to observe the realistic prospects for actual use.

6.2 Result of experiment and feature analysis

Table 8 shows the results. Though they are reasonable, the overall accuracy, especially for the partially closed test, shows that the method is in need of improvement.

In order to evaluate the effectiveness of the feature sets, we carried out experiments only using and without using each feature set. Table 9 shows that how efficient is each feature set defined in Table 6. The left-hand column in Table 9 shows the result with all feature sets except focal feature set, and the right-hand column shows the result when only the focal feature set was used.

The experiment showed that the feature set that contributed most was C (features related to the Japanese target verbs). We also carried out an experiment to check which features are effective among this set, in the same manner as the experiments for checking the effectiveness of the feature sets. The result showed that the feature $D_{modifying\_case}$ is the feature that contributed the most by far. In Japanese, case markers are strongly correlated with the voice of verbs, and the coverage of this feature for tokens related to voice is high because it is common for a verb to be modified by the case element with the case marker.

It became clear that the numeric features A and E contribute little to the overall accuracy. Table 10 shows the correlation coefficient between the numeric features and correct answers. The table shows that there is no noticeable relation between the nu-

	accuracy	(+)precision	(+)recall	(-)precision	(-)recall
Cross validation	0.646 (291/452)	0.656 (138/214)	0.614 (138/226)	0.643 (153/238)	0.677 (153/226)
Partially closed	0.521 (583/1120)	0.277 (193/697)	0.854 (193/226)	0.922 (390/423)	0.436 (390/894)

Table 8: The accuracy of classification

	without this feature set			using only this feature set		
feature set	accuracy	(+)precision	(+)recall	accuracy	(+)precision	(+)recall
(A)	0.638	0.638 (144/226)	0.639 (144/226)	0.521	0.541 (62/115)	0.277 (62/226)
(B)	0.634	0.649 (132/203)	0.584 (132/226)	0.563	0.549 (159/290)	0.705 (159/226)
(C)	0.579	0.576 (136/237)	0.604 (136/226)	0.610	0.620 (128/207)	0.570 (128/226)
(D)	0.645	0.654 (138/212)	0.615 (138/226)	0.523	0.679 (19/29)	0.087 (19/226)
(E)	0.629	0.666 (117/175)	0.518 (117/226)	0.492	0.491 (101/205)	0.447 (101/226)

Table 9: The evaluation result for each feature set

feature	MI	f(+)	f(-)
$D_{modifying\_agent}$=NIYOTTE	0.843	15	17
$E_{POS:POS\_after}$=VVN:NN	0.656	14	22
E_{POS_before}=IN	0.536	10	19
E_{POS_before}=JJ	0.530	12	23
$D_{modified\_case}$=GA	0.428	13	29
$N_{grandparent\_voice}$=passive	0.408	17	39
$N_{grandchild\_voice}$=passive	0.368	14	34
E_{POS}=VVZ	0.368	14	34
F_{suffix}=NARU	0.225	0	23
$N_{case\_adjacency}$=GA:TO	0.225	0	12
F_{suffix}=SHIMAU	0.225	0	16
E_{POS}=RB	0.225	0	10
$E_{POS:POS\_after}$=VVG:DT	0.225	0	10
$E_{POS:POS\_after}$=VVN:TO	0.179	2	42
$E_{POS:POS\_after}$=VVN:SENT	0.159	3	44
$D_{modifying\_agent}$=NI	0.154	4	54

Table 7: Features which have high correlation with positive and negative examples

feature set (A)	
$EN_{\#word}$	0.038
$EN_{\#pause}$	-0.069
$EN_{\#verb}$	-0.003
$EN_{\#VVN}$	-0.061
$EN_{\#word\_len}$	0.033
feature set (E)	
$J_{\#morpheme}$	0.083
$J_{\#pause}$	0.011
$J_{\#verb}$	0.056
$J_{\#passive}$	0.035
$J_{\#depth}$	0.098

Table 10: The correlation coefficient between each feature and correct answer

meric features and the correct results. We introduced most numeric features based on the study of readability. In readability studies, however, these features are defined in terms of the overall document, and not in terms of individual sentences or of verb phrases. It would be preferable to develop numerical features that can properly reflect the nature of individual sentences or smaller constructions.

Table 9 shows that the result when only using the feature set D has a very low recall, but the highest precision of all the feature sets. This mean that there are not many occasions on which the feature set D can be applied, but when it is applied, the result is reliable. The feature set D thus is efficient as a trigger once it is applied, and the different treatment of the tokens that contain this feature set may contribute to the performance improvement.

6.3 Diagnosis

The critical cases from the point of view of improving the performance are the false positives and false negatives. We thus manually analysed the false positives and false negatives obtained in the partially closed experiment (in the actual application environment, as much training data as available should be used; we thus used the results of the partially

closed experiment here). For the false positive, we extracted 100 sample sentences from 504 sentences. For the false negative we used all 33 sentences. We asked two translators to judge whether (i) it would be better to modify the draft translations or (ii) it would not be necessary to modify the draft translations.

6.3.1 False positives

From the 100 sample sentences, we excluded 23 cases, 18 of which were judged as in need of modification by one of the translators and 5 of which were judged as in need of modification by both of the translators. We manually analysed the remaining 77 cases. Rather than the problems with the features that we used, we identified the potential factors that would contribute to the restriction of modification. Three types of restricting factor were recognised:

1. The nature of individual verbs allows or requires the passive voice. Within the data, three subtypes were identified, i.e. (i) the use of the passive is natural irrespective of context, as in "消費され (consumed)" (48 cases); (ii) the use of the passive is natural within certain fixed syntactic patterns, as in "X と呼ばれる Y (Y called X)" (10 cases); and (iii) the passive is used as part of a common collocation, as in "不安に襲われた (attacked by anxiety)" (2 cases);
2. The use of the active voice is blocked by selectional restrictions, as in "作られた堆積物 (a sediment made by ...)" (1 case); and
3. The structure of the sentence requires the passive, as in "最大の企業はすべて車を製造する企業であり、その中で石油の大半が浪費されていた (The biggest companies were all companies making cars, in which most of the oil was consumed)" (16 cases).

Together they cover 73 cases (in 4 out of 77 cases we could not identify the factor, and in 4 of the 73 cases two of the above factors were identified). It is anticipated that the first type (60 cases; about 85%) could be dealt with by introducing "pulling" triggers, i.e. using large corpora to identify the characteristics of the use of voice for individual verbs, in order to enable the system to judge the desirability of given expressions vis-à-vis the conventional alternatives. To deal with the second type requires a detailed semantic description of nouns, which is difficult to achieve, though in some cases it could be approximated by collocational tendencies. In regards to the third type of false positive, we expected that the type of features used in the experiment would have been sufficient to eliminate them, but this was not the case. In fact, many of the features require discourse level information, such as the choice of subject within the flow of discourse, in order to function properly, which we did not take into account. Although high-performance discourse processing is still in an embryonic stage, in the setting of the present study the correspondence between key information in English and that in Japanese could be used to deal with this type of false positive.

6.3.2 False negatives

Here, it is necessary to find factors that would promote modification. Among the 33 false negatives, 4 were judged as not in need of modification by both the translators. We thus examined the remaining 29 cases. In 13 cases, the verb was replaced by another verb. Including these cases, we identified four basic factors that are related to triggering modification:

1. The nature of the individual verbs strongly requires the active voice, either independently or within the particular context, as in "から尋ねられました (was asked by)" (9 cases);
2. The structure of the sentence is rendered rather awkward by the use of passives, as in "に発表されたアナリストたちによるレポートである (a report published in ..... by analysts)" (4 cases);
3. A given lexical collocation is unnatural or awkward, as in "すべての投資がふるいにかけられることを共同で要求し (that all investments be screened is collectively insisted)" (2 cases); and
4. A lexicalised collocation in the draft was subtly awkward and there is a better collocation or expression that fits the situation (14 cases).

Together they cover 26 cases. We could not identify features in 3 cases. As in false positives, the first, second and fourth types (22 cases or about 85% are fully covered by these three types) could be dealt with by introducing "pulling" triggers, using large external corpora.

For the overall data, we would expect that around 85% of 388 (77% of 504 cases) false positives (330

cases) could be dealt with by introducing "pulling" triggers. If these false positives could be removed completely, the precision would become well over 0.5 (193/(697-330)) and the ratio of notified cases would become about one third ((697-330)/1120) of the total relevant cases. Though it is unreasonable to assume this ideal case, this indicates that the features we defined and introduced in this study — though limited to those related to "pushing" triggers — were effective, and that what we have achieved by using these features is very promising in terms of realising a system that notifies users of awkward translations.

7 Conclusions

In this paper, we examined the factors that trigger modifications when translators are revising draft translations, and identified computationally tractable features relevant to the modification. We carried out an experiment for automatic detection of modification candidates. The result was highly promising, though it revealed several issues that need to be addressed further.

Following the results reported in this paper, we are currently working on.

(i) extending the experiment by introducing outside data to carry out open experiments (we have obtained draft and final translations of three more books);

(ii) introducing the degree of necessity for modifications by asking translators to judge the data; and

(iii) further examining the features used in the experiment for the improvement of performance.

In addition, we are experimenting with a method for making use of large-scale external corpora in order to deal with "pulling"-type triggers, with additional features taken from large external corpora.

Acknowledgement

This research is partly supported by grant-in-aid (A) 17200018 "Construction of online multilingual reference tools for aiding translators" by the Japan Society for the Promotion of Sciences (JSPS), and also by grant-in-aid from The HAKUHO FOUNDATION, Tokyo.

References

Abekawa, T. and Kageura, K. 2007. A translation aid system with a stratified lookup interface. In *Proc. of ACL 2007 Demos and Poster Sessions*, p. 5–8.

Anzai, T. 1995. *Eibun Hon'yaku Jutu* (in Japanese). Tokyo: Chikuma.

Barzilay, R. and McKeown, K. R. 2001. Extracting paraphrases from a parallel corpus. In *Proc. of ACL 2001*, p. 50-57.

Barzilay, R. and Lee, L. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proc. of HLT-NAACL 2003*, p. 16-23.

Dolan, B. et. al. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources alignment. In *Proc. of COLING 2004*, p. 350-356.

Fry, E. 1968. A readability formula that saves time. *Journal of Reading*, 11, p. 513-516, 575-578.

Gunning, R. 1959. *The Technique of Clear Writing*. New York: McGraw-Hill.

Haruno, M. and Yamazaki, T. 1996. High-performance bilingual text alignment using statistical and dictionary information. In *Proc. of ACL 1996*, p. 131-138.

Inui, K. and Fujita, A. 2004. A survey on paraphrase generation and recognition. *Journal of Natural Language Processing*, 11(5), p. 131-138.

Leggett, J. 2005. *Half Gone*. London: Portobello. [Masuoka, K. et. al. trans. 2006. *Peak Oil Panic*. Tokyo: Sakuhinsha.]

Nakamura, Y. 2003. *Eiwa Hon'yaku no Genri Gihou* (in Japanese). Tokyo: Nichigai Associates.

Och, F. J. and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1), p. 19-51.

Quirk, C., Brocktt, C. and Dolan, W. B. 2004 Monolingual machine translation for paraphrase generation. In *Proc. of EMNLP 2004*, p. 142-149.

Schmid, H. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proc. of NeMLAP*, p. 44-49.

Shinyama, Y. et. al. 2002. Automatic paraphrase acquisition from news articles. In *Proc. of HLT 2002*, p. 40-46.

Sun, et. al. 2007. Detecting erroneous sentences using automatically mined sequential patterns. In *Proc. of ACL 2007*, p. 81-88.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. New York: Springer.

Improving Word Alignment by Adjusting Chinese Word Segmentation

Ming-Hong Bai[1,2] **Keh-Jiann Chen**[1] **Jason S. Chang**[2]
1 Institute of Information Science, Academia Sinica
2 Department of Computer Science, National Tsing-Hua University
mhbai@sinica.edu.tw kchen@iis.sinica.edu.tw jschang@cs.nthu.edu.tw

Abstract

Most of the current Chinese word alignment tasks often adopt word segmentation systems firstly to identify words. However, word-mismatching problems exist between languages and will degrade the performance of word alignment. In this paper, we propose two unsupervised methods to adjust word segmentation to make the tokens 1-to-1 mapping as many as possible between the corresponding sentences. The first method is learning affix rules from a bilingual terminology bank. The second method is using the concept of impurity measure motivated by the decision tree. Our experiments showed that both of the adjusting methods improve the performance of word alignment significantly.

1 Introduction

Word alignment is an important preprocessing task for statistical machine translation. There have been many statistical word alignment methods proposed since the IBM models have been introduced. Most existing methods treat word tokens as basic alignment units (Brown et al., 1993; Vogel et al., 1996; Deng and Byrne, 2005), however, many languages have no explicit word boundary markers, such as Chinese and Japanese. In these languages, word segmentation (Chen and Liu, 1992; Chen and Bai, 1998; Chen and Ma, 2002; Ma and Chen, 2003; Gao et al., 2005) is often carried out firstly to identify words before word alignment (Wu and Xia, 1994). However, the differences in lexicalization may degrade word alignment performance, for different languages may realize the same concept using different numbers of words (Ma et al., 2007; Wu, 1997). For instance, Chinese multi-syllabic words composed of more than one meaningful morpheme which may be translated to several English words. For example, the Chinese word *教育署* is composed of two meaning units, *教育* and *署*, and is translated to *Department of Education* in English. The morphemes *教育* and *署* have their own meanings and are translated to *Education* and *Department* respectively. The phenomenon of lexicalization mismatch will degrade the performance of word alignment for several reasons. The first reason is that it will reduce the cooccurrence counts of Chinese and English tokens. Consider the previous example. Since *教育署* is treated as a single unit, it does not contribute to the occurrence counts of *Education*/*教育* and *Department*/*署* token pairs. Secondly, the rarely occurring compound word may cause the *garbage collectors* effect (Moore, 2004; Liang et al., 2006), aligning a rare word in source language to too many words in the target language, due to the frequency imbalance with the corresponding translation words in English (Lee, 2004). Finally, the IBM models (Moore, 2004) impose the limitation that each word in the target sentence can be generated by at most one word in the source sentence. In this case, a many-to-one alignment, links a phrase in the source sentence to a single token in the target sentence, is not allowed, forcing most links of a phrase in the source sentence to be abolished. As in the previous example, when aligning from English to Chinese, *教育署* can only be linked to one of the English words, say *Education*, because of the limitation of the IBM model. However for remedy, many of the current word alignment methods combine the results of both alignment directions, via *intersection* or

grow-diag-final heuristic, to improve the alignment reliability (Koehn et al., 2003; Liang et al., 2006; Ayan et al., 2006; DeNero et al., 2007). However the many-to-one link limitation will undermine the reliability due to the fact that some links are not allowed in one of the directions.

In this paper, we propose two novel methods to adjust word segmentation so as to decrease the effect of lexicalization differences to improve word alignment performance. The main idea of our methods is to adjust Chinese word segmentation according to their translation derived from parallel sentences in order to make the tokens compatible to 1-to-1 mapping between the corresponding sentences. The first method is based on learning a set of affix rules from bilingual terminology bank, and adjusting the segmentation according to these affix rules when preprocessing the Chinese part of the parallel corpus. The second method is based on the so-called *impurity* measure, which was motivated by the decision tree (Duda et al., 2001).

2 Related Works

Our methods are motivated by the translation-driven segmentation method proposed by Wu (1997) to segment words in a way to improve word alignment. However, Wu's method needs a translation lexicon to filter out the links which were not in the lexicon and the result was only evaluated on the sentence pairs which were covered by the lexicon.

A word packing method has been proposed by Ma et al. (2007) to improve the word alignment task. Before carrying out word alignment, this method packs several consecutive words together when those words believed to correspond to a single word in the other language. Our basic idea is similar to this, but on the contrary, we try to unpack words which are translations of several words in the other language. Since the word packing method treats the packed consecutive words as a single token, as we mentioned in the previous section, it weakens the association strength of translation pairs of their morphemes while applying the IBM word alignment model.

A lot of morphological analysis methods have been proposed to improve the performance of word alignment for inflectional language (Lee et al., 2003; Lee, 2004; Goldwater, 2005). They proposed to split a word into a morpheme sequence of the pattern prefix*-stem-suffix* (* denotes zero or more occurrences of a morpheme). Their experiments showed that morphological analysis can improve the quality of machine translation by reducing data sparseness and by making the tokens in two languages correspond more 1-to-1. However, these segmentation methods were developed from the monolingual perspective.

3 Adjusting Word Segmentation

The goal of word segmentation adjustment is to adjust the segmentation of Chinese words such that we have as many 1-to-1 links to the English words as possible. In this task, we will face the problem of finding the proper morpheme boundaries for Chinese words. The challenge is that almost all characters of Chinese are morphemes and therefore almost every character boundary in a word could be the boundary of a morpheme, there is no simple rules to find the suitable boundaries of morphemes. Furthermore, not all meaningful morphemes need to be segmented to meet the requirement of 1-to-1 mapping. For example, *washing machine/洗衣機* can be segmented into *洗衣* and *機* corresponding to *washing* and *machine* while *heater/暖氣機* does not need, it depends on their translations.

In this paper, we have proposed two different methods to solve this problem: 1. learning affix rules from terminology bank to segment morphemes and 2. using *impurity* measure to finding the morpheme boundaries. The detail of these methods will be described in the following sections.

4 Affix Rule Method

The main idea of this method is to segment a Chinese word according to some properly designed conditional dependent affix rules. As shown in Figure 1, each rule is composed of three conditional constraints, a) affix condition, b) English word condition and c) exception condition. In the affix condition, we place a underscore on the left of a morpheme, such as *_機*, to denote a suffix and on the right, such as *副_*, to denote a prefix. The affix rules are applied to each word by checking the following three conditions:

1. The target word has the affix.

2. The English word which is the target of translation exists in the parallel sentence.
3. The target word does not contain the morphemes in the exception list (The morpheme in the exception list shows an alternative segmentation.).

If the target word satisfies all of the above conditions of any rule, then the morpheme should be separated from the word. The remaining problem will be how to derive the set of affix rules.

affix	English word	exception
_機	machine	
_機	engine	
副_	vice	
副_	deputy	副手
_業	industry	工業

Figure 1. Samples of affix rules.

4.1 Training Data

We use an unsupervised method to extract affix rules from a Chinese-English terminology bank[1]. The bilingual terminology bank a total of 1,046,058 English terms with Chinese translations in 63 categories. Among them, 60% or 629,352 terms are compounds. We take the advantage of the terminology bank, that all terminologies are 1-to-1 well translated, to find the best morpheme segmentation from ambiguous segmentations of a Chinese word according to its English counterpart. Then we extracted affix rules from the word-to-morpheme alignment results of terms and translation.

4.2 Word-to-Morpheme Alignment

The training phase of word-to-morpheme alignment is based loosely on word-to-word alignment of the IBM model 1. Instead of using Chinese words, we considered all the possible morphemes. For example, consider the task of aligning *Department of Education* and *教育署* as shown as Figure 2. We use the EM algorithm to train the translation probabilities of word-morpheme pairs based on IBM model 1.

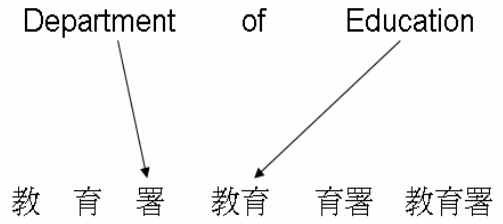


Figure 2. Example of word-to-morpheme alignment.

In the aligning phase, the original IBM model 1 does not work properly as we expected. Because the English words prefer to link to single character and it results that some correct Chinese translations will not be linked. The reason is that the probability of a morpheme, say $p(教育|education)$, is always less than its substring, $p(教|education)$, since whatever *教育* occurs *教* and *育* always occur but not vice versa. So the aligning result will be *教/Education* and *署/Department*, *育* is abandoned. To overcome this problem, a constraint of alignment is imposed to the model to ensure that the aligning result covers every Chinese characters of a target word and no overlapped characters in the result morpheme sequence. For instances, both *教/Education 署/Department* and *教育/Education 育署/Department* are not allowed alignment sequences. The constraint is applied to each possible aligning result. If the alignment violates the constraint, it will be rejected.

Since the new alignment algorithm must enumerate all of the possible alignments, the process is very time consuming. Therefore, it is advantageous to use a bilingual terminology bank rather than a parallel corpus. The average length of terminologies is short and much shorter than a typical sentence in a parallel corpus. This makes words to morphemes alignment computationally feasible and the results highly accurate (Chang et al., 2001; Bai et al., 2006). This makes it possible to use the result as pseudo gold standards to evaluate affix rules as described in section 4.3.

1 The bilingual terminology bank was compiled by the National Institute for Compilation and Translation. It is freely download at http://terms.nict.gov.tw by registering your information.

air|空氣 refrigeration|冷凍 machine|機
building|建築 industry|業
compound|複式 steam|蒸汽 engine|機
electronics|電子 industry|業
vice|副 chancellor|校長

Figure 3. Sample of word-to-morpheme alignment.

4.3 Rule Extraction

After the alignment task, we will get a word-to-morpheme aligned terminology bank as shown in Figure 3. We can subsequently extract affix rules from the aligned terminology bank by the following steps:

1) **Generate candidates of affix rule:**
   For each alignment, we produce all alignment links as affix rules. For instance, with (*electronics*| *電 子* *industry*| *業*), we would produce two rules:

   (a) *電子_, electronics*
   (b) *_業, industry*

2) **Evaluate the rules:**
   The precision of each candidate rule is estimated by applying the rule to segment the Chinese terms. If a Chinese term contains the affix shown in the rule, the affix will be segmented. The results of segmentation are then to compare with the segmentation results of the alignments done by the algorithm of the section 4.2 as pseudo gold standards. Some example results of rule evaluations are shown in Figure 4.

affix	English word	Rule Applied	Correct segments	precision
主_	master	458	378	0.825
週期_	periodic	130	100	0.769
視訊_	video	46	40	0.870
_鍊	chain	147	107	0.728
_箱	box	716	545	0.761

Figure 4. Sample evaluations of candidate rules.

3) **Adding exception condition:**
   In the third step, we sort the rules according to their precision rates in descending order, resulting in rules $R_1..R_n$. And then for each R_i, we scan R_1 to R_{i-1}, if there is a rule, R_j, have the same English word condition and the affix condition of R_i subsume that of R_j, then we add affix condition of R_j as exception condition of R_i. For example, *_業, industry* and *_工業, industry* are rule candidates in the sorted table and have the same English word condition. Furthermore, the condition *_業* subsumes that of *工業*, we add *工業* to the exception condition of the rule with a shorter affix.

4) **Reevaluate the rules with exception condition:**
   After adding the exception conditions, the rules are reevaluated with considering the exception condition to get new evaluation scores.

5) **Select rules by scores**:
   Finally, filter out the rules with scores lower than a threshold[2].

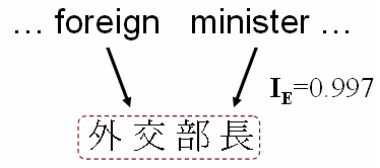
The reason of using exception condition is that an affix is usually an abbreviation of a word, such as *_業* is an abbreviation of *工業*. In general, a full morpheme is preferred to be segmented than its abbreviation while both occurred in a target word. For example, when applying rules to *電子工業 /electronic industry*, *_工業,industry* is preferred than *_業,industry*. However, in the evaluation step, precision rate of *_業,industry* will be reduced when applying to full morphemes, such as *電子工業 /electronic industry*, and then could be filtered out if the precision is lower than the threshold.

5 Impurity Measure Method

The impurity measure was used by decision tree (Duda et al., 2001) to split the training examples into smaller and smaller subsets progressively according to features and hope that all the samples in each subset is as *pure* as possible. For convenient, they define the *impurity* function rather than the *purity* function of a subset as follows:

$$impurity(S) = -\sum_j P(w_j)\log_2 P(w_j)$$

[2] We set the threshold as 0.7.



(a) impurity value of *外交部長*.

(b) impurity values of *外交* and *部長*.

Figure 5. Examples of impurity values.

Where $P(w_j)$ is the fraction of examples at set S that are in category w_j. By the well-known properties of entropy if all the examples are of the same category the impurity is 0; otherwise it is positive, with the greatest value occurring when the different classes are equal likely.

5.1 Impurity Measure of Translation

In our experiment, the impurity measure is used to split a Chinese word into two substrings and hope that all the characters in a substring are generated by the parallel English words as *pure* as possible. Here, we treat a Chinese word as a set of characters, the parallel English words as categories and the fraction of examples is redefined by the expected fraction number of characters that are generated by each English word. So we redefine the *entropy impurity* as follows:

$$I_E(f;\mathbf{e},\mathbf{f}) = -\sum_{\forall e \in \mathbf{e}} c(f \mid e;\mathbf{e},\mathbf{f}) \log_2 c(f \mid e;\mathbf{e},\mathbf{f})$$

In which f denotes the target Chinese word, **e** and **f** denote the parallel English and Chinese sentence that f belongs to and $c(f \mid e;\mathbf{e},\mathbf{f})$ is the expected fraction number of characters in f that are generated by word e. The expected fraction number can be defined as follows:

$$c(f \mid e;\mathbf{e},\mathbf{f}) = \frac{\sum_{\forall c \in f} p(c \mid e)}{\sum_{\forall e \in \mathbf{e}} \sum_{\forall c \in f} p(c \mid e)}$$

Where $p(c \mid e)$ denotes the translation probability of Chinese character c given English word e.

For example, as shown in Figure 5, the impurity value of *外交部長*, Figure 5.(a), is much higher than values of *外交* and *部長*, Figure 5.(b). Which means that the generating relations from English to Chinese tokens are purified by breaking *外交部長* into *外交* and *部長*.

The translation probabilities between Chinese characters and English word can be trained using IBM model 1 by treating Chinese characters as tokens.

5.2 Target Word Selection

In this experiment, we treat the Chinese words which can be segmented into morphemes and linked to different English words as target words. In order to speedup our impurity method only target words will be segmented during the process. Therefore we investigate the actual distribution of target words first, we have tagged 1,573 Chinese words manually with *target* and *non-target*. It turns out that only 6.87% of the Chinese words are tagged as *target* and 94.4% of target words are nouns. The results show that most of the Chinese words do not need to be re-segmented and their POS distribution is very unbalanced. The results show that we can filter out the *non-target* words by simple clues. In our experiment, we use three features to filter out *non-target* words:

1) POS: Since 94.4% of the target words are nouns, we focus our experiment on nouns and filter out words with other POS.
2) One-to-many alignment in GIZA++: Only Chinese words which are linked to multiple English words in the result of GIZA++ are considered to be target words.
3) Impurity measure: the target words are expected to have high impurity values. So the words with a impurity values larger than a threshold are selected as target words[3].

[3] In our experiment, we use 0.3 as our threshold.

5.3 Best Breaking Point

The goal of segmentation adjustment using impurity is to find the best breaking point of a Chinese word according to parallel English words. When a word is broken into two substrings, the new substrings can be compared to original word by the *information gain* which is defined in terms of impurity as follows:

$$IG(f, f_1^i, f_{i+1}^n) = I_E(f;\mathbf{e},\mathbf{f}) - \frac{1}{2} I_E(f_1^i;\mathbf{e},\mathbf{f}) - \frac{1}{2} I_E(f_{i+1}^n;\mathbf{e},\mathbf{f})$$

Where i denotes a break point in f, f_1^i denotes first i characters of f, and f_{i+1}^n denotes last n-i characters of f. If the information gain of a breaking point is positive, the result substrings are considered to be better, i.e. more pure than original word.

The goal of finding the best breaking point can be achieved by finding the point which maximizes the information gain as the following formula:

$$\arg\max_{1\le i<n} IG(f, f_1^i, f_{i+1}^n)$$

Note that a word can be separated into two substrings each time. If we want to segment a complex word composed of many morphemes, just split the word again and again like the construction of decision tree, until the information gain is negative or less than a threshold[4].

6 Experiments

In order to evaluate the effect of our methods on the word alignment task, we preprocessed parallel corpus in three ways: First we use a state-of-the-art word segmenter to tokenize the Chinese part of the corpus. Then, we used the affix rules to adjust word segmentation. Finally, we do the same but by using the impurity measure method. We used the GIZA++ package (Och and Ney, 2003) as the word alignment tool to align tokens on the three copies of preprocessed parallel corpora.

We used the first 100,000 sentences of Hong Kong News parallel corpus from LDC as our training data. And 112 randomly selected parallel sentences were aligned manually with *sure* and *possible* tags, as described in (Och and Ney, 2000), and we used these annotated data as our gold standard in testing.

Because of the modification of Chinese tokens caused by the word segmentation adjustment, a problem has been created when we wanted to compare the results to the copy which did not undergo adjustment. Therefore, after the alignment was done, we merged the alignment links related to tokens that were split up during adjustment. For example, the two links of *foreign*/外交 *minister*/部長 were merged as *foreign minister*/外交部長.

The evaluation of word alignment results are shown in Table 1, including *precision-recall* and *AER* evaluation methods. In which the *baseline* is alignment result of the unadjusted data. The table shows that after the adjustment of word segmentation, both methods obtain significant improvement over the *baseline*, especially for the English-Chinese direction and the intersection results of both directions. The *impurity* method in particular improves alignment in both English-Chinese and Chinese-English directions.

The improvement of intersection of both directions is important for machine translation. Because the intersection result has higher precision, a lot of machine translation method relies on intersecting the alignment results. The phrase-based machine translation (Koehn et al., 2003) uses the *grow-diag-final* heuristic to extend the word alignment to phrase alignment by using the intersection result. Liang (Liang et al., 2006) has proposed a symmetric word alignment model that merges two simple asymmetric models into a symmetric model by maximizing a combination of likelihood and agreement between the models. This method uses the intersection as the agreement of both models in the training time. The method has reduced the alignment error significantly over the traditional asymmetric models.

In order to analyze the adjustment results, we also manually segment and link the words of Chinese sentences to make the alignments 1-to-1 mapping as many as possible according to their translations for the 112 gold standard sentences. Table 2 shows the results of our analysis, the performance of impurity measure method is also slightly better than the affix rules in both recall and precision measure.

[4] In our experiment, we set 0 as the threshold.

	direction	Recall	precision	F-score	AER
baseline	English-Chinese	68.3	61.2	64.6	35.7
	Chinese-English	79.6	67.0	72.8	27.8
	intersection	59.9	92.0	72.6	26.6
affix rules	English-Chinese	78.2	64.6	70.8	29.8
	Chinese-English	80.2	68.0	73.6	27.0
	intersection	69.1	92.3	79.0	20.2
impurity	English-Chinese	78.1	64.9	70.9	29.7
	Chinese-English	81.4	70.4	75.5	25.0
	intersection	70.2	91.9	79.6	19.8

Table 1. Alignment results based on the standard word segmentation data.

	recall	precision
affix rules	82.35	66.66
impurity	84.31	67.72

Table 2. Alignment results based on the manual word segmentation data.

7 Conclusion

In this paper, we have proposed two Chinese word segmentation adjustment methods to improve word alignment. The first method uses the affix rules learned from a bilingual terminology bank and then applies the rules to the parallel corpus to split the compound Chinese words into morphemes according to its counterpart parallel sentence. The second method uses the impurity method, which was motivated by the method of decision tree. The experimental results show that both methods lead to significant improvement in word alignment performance.

Acknowledgements: This research was supported in part by the National Science Council of Taiwan under NSC Grants: NSC95-2422-H-001-031.

References

Necip Fazil Ayan and Bonnie J. Dorr. 2006. Going Beyond AER: An Extensive Analysis of Word Alignments and Their Impact on MT. In *Proceedings of ACL 2006*, pages 9-16, Sydney, Australia.

Ming-Hong Bai, Keh-Jiann Chen and Jason S. Chang. 2006. Sense Extraction and Disambiguation for Chinese Words from Bilingual Terminology Bank. *Computational Linguistics and Chinese Language Processing*, 11(3):223-244.

Petter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer. 1993. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263-311.

Jason S Chang, David Yu, Chun-Jun Lee. 2001. Statistical Translation Model for Phrases(in Chinese). *Computational Linguistics and Chinese Language Processing*, 6(2):43-64.

Keh-Jiann Chen, Ming-Hong Bai. 1998. Unknown Word Detection for Chinese by a Corpus-based Learning Method. *International Journal of Computational linguistics and Chinese Language Processing*, 1998, Vol.3, #1, pages 27-44.

Keh-Jiann Chen, Wei-Yun Ma. 2002. Unknown Word Extraction for Chinese Documents. In *Proceedings of COLING 2002*, pages 169-175, Taipei, Taiwan.

Keh-Jiann Chen, Shing-Huan Liu. 1992. Word Identification for Mandarin Chinese Sentences. In *Proceedings of 14th COLING*, pages 101-107.

John DeNero, Dan Klein. 2007. Tailoring Word Alignments to Syntactic Machine Translation. In *Proceedings of ACL 2007*, pages 17-24, Prague, Czech Republic.

Yonggang Deng, William Byrne. 2005. HMM word and phrase alignment for statistical machine translation. In *Proceedings of HLT-EMNLP 2005*, pages 169-176, Vancouver, Canada.

Richard O. Duda, Peter E. Hart, David G. Stork. 2001. *Pattern Classification*. John Wiley & Sons, Inc.

Jianfeng Gao, Mu Li, Andi Wu and Chang-Ning Huang. 2005. Chinese word segmentation and named entity recognition: a pragmatic approach. *Computational Linguistics,* 31(4)

Sharon Goldwater, David McClosky. 2005. Improving Statistical MT through Morphological Analysis. In

Proceedings of HLT/EMNLP 2005, pages 676-683, Vancouver, Canada.

Philipp Koehn, Franz J. Och, Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT/NAACL 2003*, pages 48-54, Edmonton, Canada.

Young-Suk Lee. 2004. Morphological Analysis for Statistical Machine Translation. In *Proceedings of HLT-NAACL 2004*, pages 57-60, Boston, USA.

Young-Suk Lee, Kishore Papineni, Salim Roukos. 2003. Language Model Based Arabic Word Segmentation. In *Proceedings of ACL 2003*, pages 399-406, Sapporo, Japan.

Percy Liang, Ben Taskar, Dan Klein. 2006. Alignment by Agreement. In *Proceedings of HLT-NAACL 2006*, pages 104-111, New York, USA.

Wei-Yun Ma, Keh-Jiann Chen. 2003. A Bottom-up Merging Algorithm for Chinese Unknown Word Extraction. In *Proceedings of ACL 2003, Second SIGHAN Workshop on Chinese Language Processing*, pp31-38, Sapporo, Japan.

Yanjun Ma, Nicolas Stroppa, Andy Way. 2007. Bootstrapping Word Alignment via Word Packing. In *Proceedings of ACL 2007*, pages 304-311, Prague, Czech Republic.

Robert C. Moore. 2004. Improving IBM Word-Alignment Model 1. In *Proceedings of ACL 2004*, pages 519-526, Barcelona, Spain.

Franz Josef Och, Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, volume 29, number 1, pp. 19-51 March 2003.

Franz J. Och, Hermann Ney., Improved Statistical Alignment Models, In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 2000, Hong Kong, pp. 440-447.

Stefan Vogel, Hermann Ney, Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of COLING 1996*, pages 836-841, Copenhagen, Denmark.

Dekai Wu, Xuanyin Xia. 1994. Learning an English-Chinese Lexicon from a Parallel Corpus. In *Proceedings of AMTA 1994*, pages 206-213, Columbia, MD.

Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377-403.

The Telling Tail: Signals of Success in Electronic Negotiation Texts

Marina Sokolova
IRO, Université de Montréal
Montréal, Québec, Canada
sokolovm@iro.umontreal.ca

Vivi Nastase
EML Research, gGmbH
Heidelberg,Germany
nastase@eml-research.de

Stan Szpakowicz
SITE, University of Ottawa
Ottawa, Ontario, Canada
ICS, Polish Academy of Sciences
Warsaw, Poland
szpak@site.uottawa.ca

Abstract

We analyze the linguistic behaviour of participants in bilateral electronic negotiations, and discover that particular language characteristics are in contrast with face-to-face negotiations. Language patterns in the later part of electronic negotiation are highly indicative of the successful or unsuccessful outcome of the process, whereas in face-to-face negotiations, the first part of the negotiation is more useful for predicting the outcome. We formulate our problem in terms of text classification on negotiation segments of different sizes. The data are represented by a variety of linguistic features that capture the gist of the discussion: negotiation- or strategy-related words. We show that, as we consider ever smaller final segments of a negotiation transcript, the negotiation-related words become more indicative of the negotiation outcome, and give predictions with higher $Accuracy$ than larger segments from the beginning of the process.

1 Introduction

We use language every day to convince, explain, manipulate and thus reach our goals. This aspect of language use is even more obvious in the context of negotiations. The parties must reach an agreement on the partitioning or sharing of a resource, while each party usually wants to leave the negotiation table with the larger piece of the pie. These tendencies become stronger when negotiators use only electronic means to communicate, that is to say, participate in electronic negotiations. In face-to-face contact, prosody and body language often have a crucial role in conveying attitudes and feelings. E-negotiators, on the other hand, must rely only on texts. We perform automatic analysis of the textual data in e-negotiations. We identify linguistic expressions of such negotiation-specific behaviour that are indicative of the final outcome of the process – success or failure – and observe how powerful a tool language is in helping people get what they want.

In this paper we focus on the negotiation as an ongoing process. We analyze the linguistic features of messages exchanged at various points in the course of the negotiation, to determine the time frame in which the outcome becomes decided. From our experimental point of view, we determine the segment of the negotiation which is most predictive of the outcome. There is an imposed three-week deadline in the electronic negotiations that we analyze. We hypothesize that the pressure of the deadline is reflected in the messages exchanged. The messages written later in the process are more indicative of the outcome of the process. Our empirical results support this hypothesis; an analysis of the linguistic features that make this prediction possible shows what the negotiators' main concerns are as the deadline draws near.

Here is what our results contribute to the field of text analysis. Research on text records of face-to-face negotiations suggests that the language patterns used in the first half of a negotiation predict the negotiation outcome better than those in the second half (Simons, 1993). The explanation was that

in the first phase people establish contact, exchange personal information and engage in general polite conversation, creating a foundation of trust between partners. No numerical data, however, supported this diagnosis, and there was no distinction between the prediction of successful and unsuccessful outcomes. When it comes to text classification, our hypothesis says that the classification of the second parts of e-negotiation texts is more accurate with respect to the outcome than the classification of the first parts. This makes e-negotiation texts different from newsgroup messages, newspaper articles and other documents classified by Blatak et al. (2004), where texts showe d better classification $Accuracy$ on their initial parts. We report the results of several sets of Machine Learning (ML) experiments. Performed on varying-size text data segments, they support our hypothesis.

We worked with a collection of transcripts of negotiations conducted over the Internet using the Web-based negotiation support system *Inspire* (Kersten and Zhang, 2003). Kersten and Zhang (2003) and Nastase (2006) classified e-negotiation outcomes using non-textual data. Classification based on texts is discussed in (Sokolova et al, 2005; Sokolova and Szpakowicz, 2006). None of those experiments considered segmenting the data, although Sokolova and Szpakowicz (2006) analyzed the importance of the first part of e-negotiations. The work we present here is the first attempt to investigate the effect of parts of e-negotiation textual data on classification quality. In this study we do not report types of expressions that are relevant to success and failure of negotiations. These expressions have been presented and analyzed in (Sokolova and Szpakowicz, 2005).

In section 2 we take a brief look at other work on the connection between behaviour and language. In section 3 we present our data and their representation for ML experiments, and we further motivate our work. Section 4 describes the experiments. We discuss the results in Section 5. Section 6 draws conclusions and discusses a few ideas for future work.

2 Background Review

Young (1991) discusses the theory that the situation in which language is used affects the way in which it is used. This theory was illustrated with a particular example of academic speech.

The field of neuro-linguistic programming investigates how to program our language (among other things) to achieve a goal. In the 1980s, Rodger Bailey developed the Language and Behaviour Profile based on 60 meta-programs. Charvet (1997) presents a simplified approach with 14 meta-programs. This profile proposes that people's language patterns are indicators of behavioural preferences. In the study of planning dialogues (Chu-Carroll and Carberry, 2000), Searle's theory of speech acts used through the discourse analysis also supports the fact that language carries much of people's behaviour and emotions. Reitter and Moore (2007) studied repetitions in task-oriented conversations. They demonstrated that a speaker's short-term ability to copy the interlocutor's syntax is autonomous from the success of the task, whereas long-term adaptation varies with such success.

We consider a negotiation to be a communication in which the participants want to reach an agreement relative to the splitting/sharing of resources. Language is one of the tools used to reach the goal. We propose that not all messages exchanged throughout a negotiation have the same effect on the negotiation outcome. To test this hypothesis, we take an ever smaller segment of the negotiation, and see how well we can predict the outcome of the process, based only on the messages in this fragment.

We encountered several challenges in predicting e-negotiation outcomes using the messages exchanged. First, electronic negotiations usually do not have a sequential-stage model of behaviour (Koeszegi et al, 2007), which is common in face-to-face negotiations (Adair and Brett, 2005). Here is an example of behavioural phases in face-to-face negotiations: *Perform Relational Positioning* → *Identify the Problem* → *Generate Solutions* → *Reach Agreement*. Unexpected turns and moves – typical of human behaviour – make prediction of the negotiation outcome difficult. In case of electronic negotiation, the absence of the usual negotiation structure further complicates the outcome prediction. This distinguishes e-negotiations from agent-customer phone conversations studied in (Takeuchi et al, 2007), where an agent follows the call flow pre-defined by his company's policy.

The longer an e-negotiation takes, the more elaborate the structure of the e-negotiation process becomes. Simpler e-negotiation may involve an exchange of well-structured business documents such as pre-defined contract or retail transactions. A more complex process comprises numerous offers and counter-offers and has a high degree of uncertainty because of the possible unpredictability of negotiation moves.

The next challenge stems from the limitations imposed by the use of electronic means. This overloads text messages with various tasks: negotiation issues themselves, introductions and closures traditional in negotiations, and even socializing. On the other hand, electronic means make the contacts less formal, allowing people to communicate more freely. As a result, the data have a high volume of informality such as abbreviations or slang.

The last challenge is specific to text analysis. E-negotiations usually involve a multi-cultural audience of varied background, many of whom are not native English speakers. While communicating in English, they introduce a fair amount of spelling and grammatical mistakes.

3 Textual Data in Electronic Negotiations

Participants in a negotiation assume well-defined roles (such as buyers/sellers in some business negotiations, or facilitators in legal disputes), have goals, and adopt specific behaviour to achieve those goals (Koeszegi et al, 2007). These circumstances are reflected in the language of texts exchanged in negotiations, and distinguish this type of texts from casual e-mail exchange and postings on discussion groups and chat boards. We claim that the language captured in e-negotiation textual data changes as a negotiation progresses, and that this is clearly detectable, even though it does not follow a sequential-stage model common in face-to-face-negotiations (Adair and Brett, 2005) or an agent-customer interaction call flow recommended by a company (Takeuchi et al, 2007). To support the language change hypothesis, we have conducted a series of ML experiments on negotiation segments of varying size and position, using the largest available data of electronic negotiations.

Our data come from the Web-based negotiation support system *Inspire*. *Inspire* has been used in business courses to teach students about e-negotiations and give them a chance to practice bilateral business negotiations conducted in a lightly controlled environment. For many users, conducting negotiations has been a business/ course assignment. Other users wanted to develop their English skills by participating in an *Inspire*-enabled negotiation. A negotiation would last up to three weeks, after which, if an agreement has not been reached, the systems would terminate the negotiation and record it as unsuccessful. The following is an example of a negotiation message (with the original spelling):

> *Dear Georg, I hope you are doig well. I send you this message to ask you what happened to our offer. Just be aware that we will not be indifinitely waiting on your response. As I told you during our last meeting, Itex Manufacturing needs a partnership. So it is important to me to know if you are ready to negotiate with us. We can not afford losing so much precious time. We give you now five more days to answer our offer (1st of december 1997, 2400 Swiss time). After this dead line, will propose our services to your concurrence. I still believe in a good partnership and relationship between our two societies. Let me know if you think so. For Itex Manufacturing. Rookie.*

Among the wealth of data gathered by *Inspire*, we have focussed on the accompanying text messages, extracted from the transcripts of 2557 negotiations. Each negotiation had two different participants, and one person participated in only one negotiation. The total number of contributors was over 5000; most of them were not native English speakers. The data contain $1,514,623$ word tokens and $27,055$ types. Compared with benchmark corpora, for example the Brown or the Wall Street Journal corpus (Francis and Kucera, 1997; Paul and Baker, 1992), this collection has a lower type-token ratio and a higher presence of content words among the most frequent words (this is typical of texts on a specific topic), and a high frequency of singular first- and second-person pronouns (this is typical of dialogues).

We considered all messages from one negotiation to be a single negotiation text. We concatenated the messages in chronological order, keeping the punctuation and spelling unedited. Each negotiation had a unique label, either positive or negative, and was a training example in one of two classes – success-

Features	Split	NB				SVM				DT			
		Acc	*F*	*P*	*R*	*Acc*	*F*	*P*	*R*	*Acc*	*F*	*P*	*R*
negotiation-related	1/2 and 1/2	68.1	70.4	73.0	68.0	73.6	76.8	75.4	78.2	73.9	78.8	72.1	86.8
negotiation-related	3/4 and 1/4	69.1	71.3	74.1	68.7	73.7	77.0	75.5	78.5	75.4	79.4	73.8	86.0

Table 1: *Accuracy* and corresponding $F - score$, *Precision* and *Recall*. Classifying all negotiations as successful gives a baseline *Accuracy* of 55%.

ful or unsuccessful. *Inspire* assigned a negotiation to the right class automatically. 55% of negotiations in our data set were successful, i.e. ended up with agreement.

We represented a complete negotiation, or text as we consider it, as a combined bag of words. We matched the tokens in the messages with an inventory of domains from Longman Dictionary of Contemporary English (Procter, 1978). This allowed us to select those terms that refer to negotiation specific issues – we call them *negotiation-related words*. We select *strategic words* based on words and patterns that literature shows to express the intentions, influence, self-obligations and motivations of the negotiation participants. In classifying successful and unsuccessful negotiations, subsets of these two types of features provided better *Accuracy* than statistically selected features, e.g. most frequent unigrams and unigrams with a higher log-likelihood values calculated between positive and negative classes (Sokolova et al, 2005).

We halved each text, that is to say, the complete record of a negotiation. For each half we built a bag of 123 negotiation-related words – more on this in section 4. The binary attributes represented the presence or absence of the word in its half of the text. We concatenated the two bags, and labelled the resulting bag by the outcome of the whole negotiation: positive if the negotiation was successful, negative otherwise. We repeated this procedure for the split of the negotiation text into $\frac{3}{4}$ and $\frac{1}{4}$. Our ML tools were Weka's (Witten and Frank, 2005) NAIVE BAYES (NB), the sequential minimal optimization (SVM) version of SUPPORT VECTOR MACHINE, and DECISION TREE (DT). In Table 1 we report *Accuracy* and *Precision* (P), *Recall* (R) and $F - score$ (F). P, R, F are calculated on the positive class. For every classifier, the best *Accuracy* and corresponding P, R, F are reporte d; we performed an exhaustive search on adjustable parameters; the evaluation method was tenfold cross-validation. Our *Accuracy* results are comparable with those reported in previous studies (Kersten and Zhang, 2003; Nastase, 2006; Sokolova and Szpakowicz, 2006).

We used the paired *t-test* to generalize the results on both splits.[1] The two-tailed P value was 0.0102. By conventional criteria, this difference is considered to be statistically significant.

Accuracy and, especially, *Precision* results show that DECISION TREE is sensitive to the positions of words in different parts of the negotiations. SUPPORT VECTOR MACHINE and NAIVE BAYES change *Accuracy* only slightly. The *Precision* and *Recall* results give a better picture of the performance. The presence/absence of words recorded for different splits of negotiations influences the identification of true positive examples (successful negotiations) and true negative examples (unsuccessful negotiations). *Recall* displays that DT classifies successful negotiations better when the negotiations are split $\frac{1}{2}$ and $\frac{1}{2}$. *Precision* and *Recall* together imply that unsuccessful negotiations have a higher rate of true classification achieved by NB, when the split is $\frac{3}{4}$ and $\frac{1}{4}$. This split lets us improve the worst rates of true classifications – unsuccessful negotiations for DT and successful negotiations for NB. Generally, the unequal split al lows us to reduce the difference between true positive and true negative classification results, and thus makes the classification of negotiations more balanced than the equal split. For all the three classifiers, *Accuracy* and $F - score$ are better on the $\frac{3}{4}$ and $\frac{1}{4}$ split.

4 The Empirical Set-up

We wanted to determine the placement of the segment of a negotiation most important in deciding whether the outcome is positive: at the beginning or at the end of the process. To do that, we split each negotiation in half, and built two parallel data sets, corresponding to the two halves. We classified each

[1] Results on the same data require the paired version of *t-test*.

part using various ML tools. Next, we repeated the same classification tasks using smaller and smaller final segments, in order to monitor the variation in performance. Thus each negotiation text N consisted of the head segment (h) and the tail segment (t): $N = h \bigcup t, h \bigcap t = \emptyset$, where $|t| = \frac{|N|}{i}$ and t was the segment at the end of N, and $|h| = \frac{(i-1)|N|}{i}$ covering the beginning of the negotiation. We stopped when for two consecutive splits two classifiers had better $Accuracy$ on the head than on the tail. Each segment got the same class label as the whole negotiation.

For these experiments, as briefly explained in section 3, we took the textual negotiation data represented as bags of words. Because of the large number of word features ($27,055$ tokens), we performed lexical feature selection.

Statistical analysis of the corpus built from the *Inspire* negotiation messages has revealed that the issues discussed in these messages can be grouped into a small set of topics. The particular topic or domain to which a word belongs derives from the most frequent bigram and trigram meanings; for instance, the second most frequent trigram with the word *delivery* is *payment upon delivery*, so we assign *delivery* to the domain *negotiation process*. The data come from negotiations on a specific topic (sale/purchase of bicycle parts), so a likely candidate subset would be words related to it. We select such negotiation-related words as the first set of features. We show a text sample with the negotiation-related words in **bold**:

Dear Georg, I hope you are doig well. I send you this message to ask you what happened to our **offer**. *Just be aware that we will not be indifinitely waiting on your response. As I told you during our last* **meeting**, *Itex Manufacturing needs a partnership. So it is important to me to know if you are ready to* **negotiate** *with us. We can not* **afford** *losing so much precious time. We give you now five more* **days** *to answer our* **offer** *(1st of december 1997, 2400 Swiss time). After this dead line, we will propose our services to your concurrence. I still believe in a good partnership and relationship between our two societies. Let me know if you think so. For Itex Manufacturing. Rookie.*

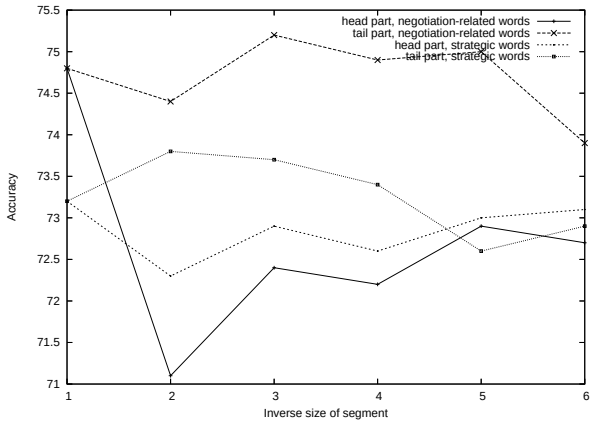
Strategies which the negotiators adopt (promises, threats, exchange of information, argumentation, and so on) affect the outcome (Sokolova and Szpakowicz, 2006). Since the messages are dense, short and grammatically simple, the expression of strategies through language is straightforward and concentrates on communicating the main goal. The word categories that convey negotiators' strategies are modals, personal pronouns, volition verbs, mental verbs; we refer to them as *strategic* words. Strategic words constitute the second set of features. Our text sample with strategic words in **bold** looks as follows:

Dear Georg, **I** *hope* **you** *are doig well.* **I** *send* **you** *this message to ask* **you** *what happened to* **our** *offer. Just be aware that* **we will not be** *indifinitely waiting on* **your** *response. As* **I** *told* **you** *during* **our last** *meeting, Itex Manufacturing needs a partnership. So it is important to* **me** *to* **know** *if* **you** *are ready to negotiate with* **us**. **We can not afford** *losing so much precious time.* **We** *give* **you** *now five more days to answer* **our** *offer (1st of december 1997, 2400 Swiss time). After this dead line,* **we will propose our** *services to* **your** *concurrence.* **I** *still* **believe** *in a good partnership and relationship between* **our** *two societies. Let* **me know** *if* **you think** *so. For Itex Manufacturing. Rookie.*

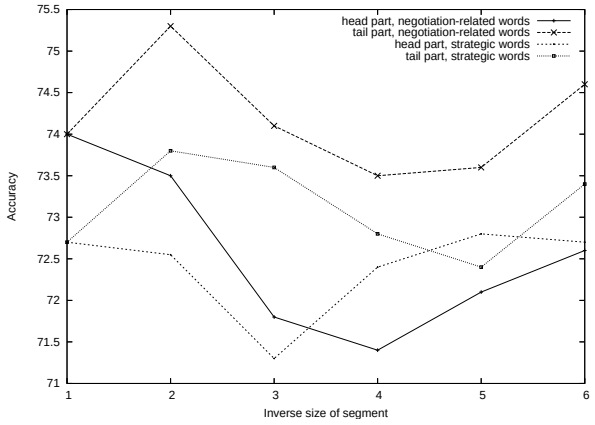
We work with kernel (SVM), decision-based (DT) and probabilistic (NB) classifiers. Applying classifiers with different working paradigms allow us to capture and understand different aspects of the data, as the results and our discussion in section 5 will show. For each classifier, we used tenfold cross-validation and exhaustive search on adjustable parameters in model selection. The best results, in particular with high overall $Accuracy$, appear in Figure 1.

When the data are represented using negotiation-related words, the *tail* segments give more accurate outcome classification than the *head* segments. This holds for all splits and all classifiers; see Figure 1. The increase in $Accuracy$ when the head segments grow was to be expected, although it does not happen with DT and SVM – only with NB. At the same time, there is no monotonic decline in $Accuracy$ when the length of the tail segments decreases. On the contrary, NB constantly improves the $Accuracy$ of the classification. We note the fact that NB increases the $Accuracy$ on both head and tail segments and makes the basic assumption of the conditional independence of features. We explain the NB results by the decreased dependence between the pres-

1. DT



2. SVM



3. NB

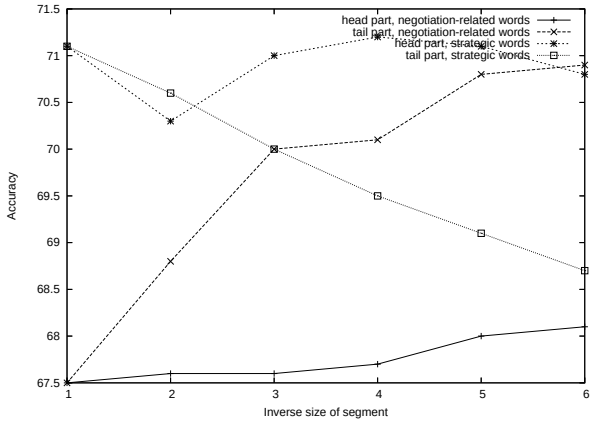


Figure 1: The classification $Accuracy$ with DT, SVM and NB, for negotiation-related and strategic words.

ence/absence of negotiation-related words when the negotiations move to the second part of the process.

The results on the strategic-word representation are slightly different for the three classifiers; see

Classifier	tail	s_1	s_2	s_1	s_2	s_3
DT	74.4	71.9	74.9	72.5	71.9	73.9
SVM	75.3	70.5	73.5	70.8	69.9	74.6
NB	68.8	68.5	70.1	68.7	68.9	70.9
Negotiation-related words						
Classifier	tail	s_1	s_2	s_1	s_2	s_3
DT	73.8	73.8	73.4	71.7	71.4	72.9
SVM	73.8	70.9	72.8	72.0	71.3	73.4
NB	60.8	70.6	69.5	69.2	69.3	68.7
Strategic words						

Table 2: The $Accuracy$ of the negotiation outcome classification on 2 and 3 splits of the second half of the negotiation – the *tail* segment. Classifying all negotiations as successful gives a baseline $Accuracy$ of 55%.

Figure 1. SVM classifies all tail segments better than head segments, DT classifies tail segments better than head segments up to the $\frac{4}{5}/\frac{1}{5}$ split, and NB classifies the tail segment better than the head segment only for the half-and-half split. The $Accuracy$ results are unstable for all three classifiers, with the $Accuracy$ on the head segments decreasing when the segments grow and the $Accuracy$ on the tail segments increasing when the tail segments shrink. The performance of the classifiers indicate that, as the deadline approaches, negotiation-related words reflect the negotiation process better than strategic words.

To investigate which part of the tail segments is more important for classifying the outcomes, we introduced additional splits in the tail segments. We divided the second half of each text into 2 and 3 parts and repeated the classification procedures for every new split. The results appear at the top of Table 2, where *tail* shows the classification results when the second half of the text was classified, and the other columns report the results on the tail splits; both splits satisfy the conditions $tail = \bigcup_i s_i$, where $s_i \bigcap s_j = \emptyset$ for every $i \neq j$.

The results show that adding splits in the *tail* segments emphasizes the importance of the last part of a negotiation. For negotiation-related word representation, the classification of the outcome on the last part of the *tail* is more accurate than on its other parts. This holds for all three classifiers. For the strategic-word representation the same is true for SVM and partially for DT, but not for NB; see the bottom of Table 2. NB classifies the negotiation outcomes more accurately on s_1 than on s_2 and on s_2 rather than s_3.

Classifier	1/3		1/4		1/5		1/6		1/7		1/8		1/9	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R
DT	74.2	85.3	74.2	84.3	75.2	82.3	73.61	83.0	74.5	82.4	72.1	81.6	74.0	81.3
SVM	76.1	78.1	76.3	76.3	77.0	75.3	78.3	75.3	77.2	73.4	76.9	72.3	77.6	71.6
NB	73.8	71.8	71.8	73.9	74.8	71.9	74.9	72.0	71.3	72.2	70.8	72.5	70.5	74.3

Table 3: $Precision$ and $Recall$ on the *tail* segments; negotiation-related words. $Precision$ and $Recall$ are calculated on the positive class.

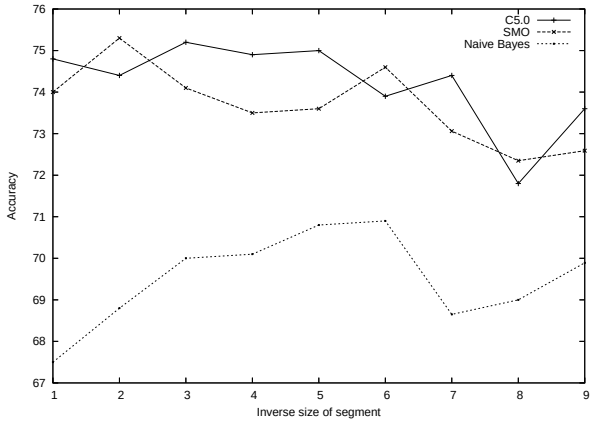


Figure 2: The evolution of the success and failure classification $Accuracy$ with decreasing segment sizes.

5 Segmentation Results

Taking into account the results reported in section 4, we chose negotiation-related words as the feature set. We selected for further analysis the half that performed better for a majority of the tools used. We focussed on the last part of the negotiation, and we extracted a gradually smaller fragment ($\frac{1}{2} - \frac{1}{9}$; 9 is the average number of text messages in one negotiation). Figure 2 plots the results of the experiments performed with decreasing segment sizes. As we see, the *tail* segment of the length $\frac{1}{7}$ gave a decline of the $Accuracy$ for SVM and NB, with a slight improvement on smaller *tail* segments.

A more detailed analysis comes from considering the $Precision$ and $Recall$ results on the segments; see Table 3. On $\frac{1}{7}$ and $\frac{1}{9}$ *tail* segments a higher $Precision$ indicates that all classifiers have improved the identification of true negatives (unsuccessful negotiations). This means that the trends in the class of unsuccessful negotiations become more noticeable for the classifiers when the deadline approaches. The $\frac{1}{8}$ split is an exception, with the abrupt drop of true negative classification by DECISION TREE. The correct classification of positive examples (successful negotiations), however, diminishes when splits become smaller; this applies to the performance of all three classifiers. This means that at the end of the negotiations the class of successful negotiations becomes more diverse and, subsequently, multi-modal, and the trends are more difficult to capture by the classifiers.

As in the previous experiments, NB's $Accuracy$ on the tail segments is higher than on the complete data. The opposite is true for SVM and DT: their $Accuracy$ on the *tail* segments is lower than on the complete data. We explain this by the fact that the sizes of *tail* segments in the last splits do not give these two classifiers sufficient information.

6 Discussion and Future Work

We have analyzed textual messages exchanged in the course of electronic negotiations. The results support our hypothesis that texts of electronic negotiation have different characteristics than records of face-to-face negotiation. In particular, messages exchanged later in the process are more informative with regard to the negotiation outcome than messages exchanged at the beginning.

We represented textual records of negotiations by two types of word features. These features capture the important aspects of the negotiation process – negotiation-related concepts and indicators of the strategies employed. We performed extensive experiments with different types of ML algorithms and segments of varying sizes from the beginning and the end of the negotiation, on a collection of over 2500 electronic negotiations. Our study shows that words expressing negotiation-related concepts are more useful for distinguishing successful and failed negotiations, especially towards the end of negotiations. We also have shown that there is no linear dependency between the segment sizes and $Accuracy$ of classification of the negotiation success and failure.

Our research plans include a continuation of the investigation of the negotiators' behaviour in electronic negotiations and its reflection in language. To see whether dialogue analysis improves prediction of the negotiation outcomes, we will look at negotiations as dialogues between participants and take into account their roles, e.g. buyer and seller. We will split a negotiation at message boundaries to avoid arbitrary splits of the negotiation process.

Acknowledgments

Partial support for this work came from the Natural Sciences and Engineering Research Council of Canada.

References

W. Adair, J. M. Brett. 2005. The negotiation dance: time, culture, and behavioral sequences in negotiation. *Organization Science*. 16(1): 33-51.

J. Blaták, E. Mráková, L. Popelínsky. 2004. Fragments and Text Categorization. *Proceedings of the 42th Annual Meeting of the Association of Computational Linguistics (ACL 2004)*. Companion Volume: 226-229, Association for Computational Linguistics.

J. M. Brett. 2001. *Negotiating Globally*. Jossey-Bass.

S. R. Charvet. 1997. *Words that Change Minds: Mastering the Language of Influence*. Kendall/Hunt.

J. Chu-Carroll, S. Carberry. 2000. Conflict Resolution in Collaborative Planning Dialogues. *International Journal of HumanComputer Studies*. 53(6): 969-1015.

W. N. Francis, H. Kučera. 1967. *Computational Analysis of Present-Day American English*, Brown University Press.

G. E. Kersten, G. Zhang. 2003. Mining Inspire Data for the Determinants of Successful Internet Negotiations. *Central European Journal of Operational Research*. 11(3): 297-316.

S. Koeszegi, E.-M. Pesendorfer, R. Vetschera. 2007. Data-driven Episodic Phase Analysis of E-negotiation. *Group Decision and Negotiation 2007*. 2: 11–130.

V. Nastase. 2006. Concession curve analysis for Inspire negotiations. *Group Decision and Negotiation*. 15: 18–193.

D. B. Paul and J. M. Baker 1992 The Design for the Wall Street Journal-based CSR Corpus. *Proceedings of the 2nd International Conference on Spoken Language Processing (ICSLP'92)*, 357-361.

P. Procter. 1978. *Longman Dictionary of Contemporary English.* Longman Group Ltd. Essex, UK.

D. Reitter, J. Moore. 2007. Predicting Success in Dialogue. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, 808-815, Association for Computational Linguistics.

T. Simons. 1993. Speech Patterns and the Concept of Utility in Cognitive Maps: the Case of Integrative Bargaining. *Academy of Management Journal*. 38(1): 139-156.

M. Sokolova, V. Nastase, M. Shah, S. Szpakowicz. 2005. Feature Selection in Electronic Negotiation Texts. *Proceedings of Recent Advances in Natural Language Processing (RANLP'2005)*, 518-524, Incoma Ltd, Bulgaria.

M. Sokolova, S. Szpakowicz. 2006. Language Patterns in the Learning of Strategies from Negotiation Texts. *Proceedings of the 19th Canadian Conference on Artificial Intelligence (AI'2006)*, 288-299, Springer.

M. Sokolova and S. Szpakowicz. 2005 Analysis and Classification of Strategies in Electronic Negotiations. *Proceedings of the 18th Canadian Conference on Artificial Intelligence (AI'2005)*, 145-157, Springer.

, H. Takeuchi, L. Subramaniam, T. Nasukawa, S. Roy. 2007 Automatic Identification of Important Segments and Expressions for Mining of Business-Oriented Conversations at Contact Centers. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 458–467, Association for Computational Linguistics.

I. Witten, E. Frank. 2005. *Data Mining, 2nd ed.*. Morgan Kaufmann. www.cs.waikato.ac.nz/ml/weka/

L. Young. 1991. *Language as Behaviour, Language as Code: A Study of Academic English*. John Benjamins.

Automatic Extraction of Briefing Templates

Dipanjan Das **Mohit Kumar** **Alexander I. Rudnicky**
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
{dipanjan, mohitkum, air}@cs.cmu.edu

Abstract

An approach to solving the problem of automatic briefing generation from non-textual events can be segmenting the task into two major steps, namely, extraction of briefing templates and learning aggregators that collate information from events and automatically fill up the templates. In this paper, we describe two novel unsupervised approaches for extracting briefing templates from human written reports. Since the problem is non-standard, we define our own criteria for evaluating the approaches and demonstrate that both approaches are effective in extracting domain relevant templates with promising accuracies.

1 Introduction

Automated briefing generation from non-textual events is an unsolved problem that currently lacks a standard approach in the NLP community. Broadly, it intersects the problem of language generation from structured data and summarization. The problem is relevant in several domains where the user has to repeatedly write reports based on events in the domain, for example, weather reports (Reiter et al., 2005), medical reports (Elhadad et al., 2005), weekly class project reports (Kumar et al., 2007) and so forth. On observing the data from these domains, we notice a *templatized* nature of report items. Examples (1)-(3) demonstrate equivalents in a particular domain (Reiter et al., 2005).

(1) [A warm front] from [Iceland] to [northern Scotland] will move [SE] across [the northern North Sea] [today and tomorrow]

(2) [A warm front] from [Iceland] to [the Faeroes] will move [ENE] across [the Norwegian Sea] [this evening]

(3) [A ridge] from [the British Isles] to [Iceland] will move [NE] across [the North Sea] [today]

In each sentence, the phrases in square brackets at the same relative positions form the slots that take up different values at different occasions. The corresponding template is shown in (4) with slots containing their respective domain entity types. Instantiations of (4) may produce (1)-(3) and similar sentences. This kind of sentence structure motivates an approach of segmenting the problem of closed domain summarization into two major steps of automatic template extraction and learning aggregators, which are pattern detectors that assimilate information from the events, to populate these templates.

(4) [PRESSURE ENTITY] from [LOCATION] to [LOCATION] will move [DIRECTION] across [LOCATION] [TIME]

In the current work we address the first problem of automatically extracting domain templates from human written reports. We take a two-step approach to the problem; first, we cluster report sentences based on similarity and second, we extract template(s) corresponding to each cluster by aligning the instances in the cluster. We experimented with two independent, arguably complementary techniques for clustering and aligning – a predicate argument based approach that extracts more general templates containing one predicate and a ROUGE (Lin, 2004) based

approach that can extract templates containing multiple verbs. As we will see below, both approaches show promise.

2 Related Work

There has been instances of template based summarization in popular Information Extraction (IE) evaluations like MUC (Marsh & Perzanowski, 1998; Onyshkevych, 1994) and ACE (ACE, 2007) where hand engineered slots were to be filled for events in text; but the focus lay on template filling rather than their creation. (Riloff, 1996) describes an interesting work on the generation of extraction patterns from untagged text, but the analysis is syntactic and the patterns do not resemble the templates that we aim to extract. (Yangarber et al., 2000) describe another system called ExDisco, that extracts event patterns from un-annotated text starting from seed patterns. Once again, the text analysis is not deep and the patterns extracted are not sentence surface forms.

(Collier, 1998) proposed automatic domain template extraction for IE purposes where MUC type templates for particular types of events were constructed. The method relies on the idea from (Luhn, 1958) where statistically significant words of a corpus were extracted. Based on these words, sentences containing them were chosen and aligned using subject-object-verb patterns. However, this method did not look at arbitrary syntactic patterns.

(Filatova et al., 2006) improved the paradigm by looking at the most frequent verbs occurring in a corpus and aligning subtrees containing the verb, by using the syntactic parses as a similarity metric. However, long distance dependencies of verbs with constituents were not looked at and deep semantic analysis was not performed on the sentences to find out similar verb subcategorization frames. In contrast, in our predicate argument based approach we look into deeper semantic structures, and align sentences not only based on similar syntactic parses, but also based on the constituents' roles with respect to the main predicate. Also, they relied on typical Named Entities (NEs) like location, organization, person etc. and included another entity that they termed as NUMBER. However, for specific domains like weather forecasts, medical reports or student reports, more varied domain entities form slots in templates, as we observe in our data; hence, existence of a module handling domain specific entities become essential for such a task. (Surdeanu et al., 2003) identify arguments for predicates in a sentence and emphasize how semantic role information may assist in IE related tasks, but their primary focus remained on the extraction of PropBank (Kingsbury et al., 2002) type semantic roles.

To our knowledge, the ROUGE metric has not been used for automatic extraction of templates.

3 The Data

3.1 Data Description

Since our focus is on creating summary items from events or structured data rather than from text, we used a corpus from the domain of weather forecasts (Reiter et al., 2005). This is a freely available parallel corpus[1] consisting of weather data and human written forecasts describing them. The dataset showed regularity in sentence structure and belonged to a closed domain, making the variations in surface forms more constrained than completely free text. After sentence segmentation we arrived at a set of 3262 sentences. From this set, we selected 3000 for template extraction and kept aside 262 sentences for testing.

3.2 Preprocessing

For semantic analysis, we used the ASSERT toolkit (Pradhan et al., 2004) that produces shallow semantic parses using the PropBank conventions. As a by product, it also produces syntactic parses of sentences, using the Charniak parser (Charniak, 2001). For each sentence, we maintained a part-of-speech tagged (leaves of the parse tree), parsed, baseNP[2] tagged and semantic role tagged version. The baseNPs were retrieved by pruning the parse trees and not by using a separate NP chunker. The reason for having a baseNP tagged corpus will become clear as we go into the detail of our template extraction techniques. Figure 1 shows a typical output from the Charniak parser and Figure 2 shows the same tree with nodes under the baseNPs pruned.

We identified the need to have a domain entity tagger for matching constituents in the sentences.

[1] http://www.csd.abdn.ac.uk/research/sumtime/

[2] A baseNP is a noun-phrase with no internal noun-phrase

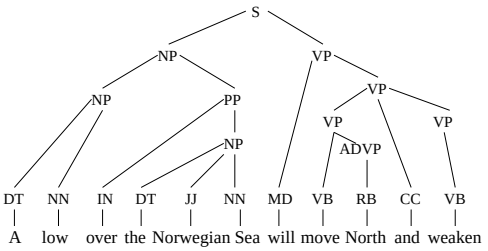


Figure 1: Parse tree for a sentence in the data.

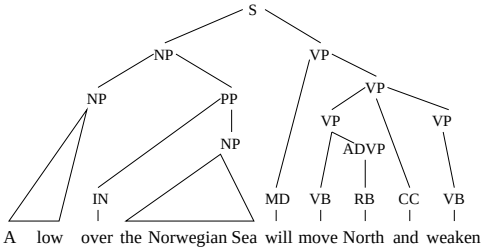


Figure 2: Pruned parse tree for a sentence in the corpus

Any tagger for named entities was not suitable for weather forecasts since unique constituent types assumed significance unlike newswire data. Since the development of such a tagger was beyond the scope of the present work, we developed a module that took baseNP tagged sentences as input and produced tags across words and baseNPs that were domain entities. The development of such a module by hand was easy because of a limited vocabulary ($<$ 1000 words) of the data and the closed set nature of most entity types (e.g the *direction* entity could take up a finite set of values). From inspection, thirteen distinct entity types were recognized in the domain. Figure 3 shows an example output from the entity recognizer with the sentence from Figure 2 as input.

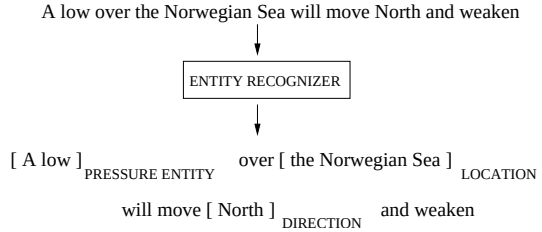


Figure 3: Example output of the entity recognizer

We now provide a detailed description of our clustering and template extraction algorithms.

4 Approach and Experiments

We adopted two parallel approaches. First, we investigated a predicate-argument based approach where we consider the set of all propositions in our dataset, and cluster them based on their verb subcategorization frame. Second, we used ROUGE, a summarization evaluation metric that is generally used to compare machine generated and human written summaries. We uniquely used this metric for clustering similar summary items, after abstracting the surface forms to a representation that facilitates comparison of a pair of sentences. The following subsections detail both the techniques.

4.1 A Predicate-Argument Based Approach

Analysis of predicate-argument structures seemed appropriate for template extraction for a few reasons: Firstly, complicated sentences with multiple verbs are broken down into propositions by a semantic role labeler. The propositions[3] are better generalizable units than whole sentences across a corpus. Secondly, long distance dependencies of constituents with a particular verb, are captured well by a semantic role labeler. Finally, if verbs are considered to be the center of events, then groups of sentences with the same semantic role sequences seemed to form clusters conveying similar meaning. We explain the complete algorithm for template extraction in the following subsections.

(5) `[ARG0 A low over the Norwegian Sea] [AGM-MOD will] [TARGET move ] [ARGM-DIR North ] and weaken`

(6) `[ARG0 A high pressure area ] [AGM-MOD will ] [TARGET move] [ARGM-DIR southwestwards] and build on Sunday.`

4.1.1 Verb based clustering

We performed a verb based clustering as the first step. Instead of considering a unique set of verbs, we considered related verbs as a single verb type. The relatedness of verbs was derived from Wordnet (Fellbaum, 1998), by merging verbs that appear in the same synset. This kind of clustering is not

[3] sentence fragments with one verb

ideal in a corpus containing a huge variation in event streams, like newswire. However, the results were good for the weather domain where the number of verbs used is limited. The grouping procedure resulted in a set of 82 clusters with 6632 propositions.

4.1.2 Matching Role Sequences

Each verb cluster was considered next. Instead of finding structural similarities of the propositions in one go, we first considered the semantic role sequences for each proposition. We searched for propositions that had exactly similar role sequences and grouped them together. To give an example, both sentences 5 and 6 have the matching role sequence ARG0–ARGM-MOD–TARGET–ARGM-DIR. The intuition behind such clustering is straightforward. Propositions with a matching verb type with the same set of roles arranged in a similar fashion would convey similar meaning. We observed that this was indeed true for sentences tagged with correct semantic role labels.

Instead of considering matching role sequences for a set of propositions, we could as well have considered matching bag of roles. However, for the present corpus, we decided to use strict role sequence instead because of the sentences' rigid structure and absence of any passive sentences. This subclustering step resulted in smaller clusters, and many of them contained a single proposition. We threw out these clusters on the assumption that the human summarizers did not necessarily have a template in mind while writing those summary items. As a result, many verb types were eliminated and only 33 verb-type clusters containing several subclusters each were produced.

4.1.3 Looking inside Roles

Groups of propositions with the same verb-type and semantic role sequences were considered in this step. For each group, we looked at individual semantic roles to find out similarity between them. We decided at first to look at syntactic parse tree similarities between constituents. However, there is a need to decide at what level of abstraction should one consider matching the parse trees. After considerable speculation, we decided on pruning the constituents' parse trees till the level of baseNPs and then match the resulting tag sequences.

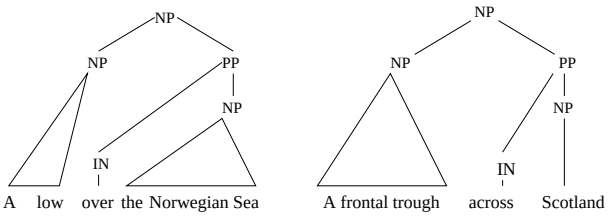


Figure 4: Matching ARG0s for two propositions

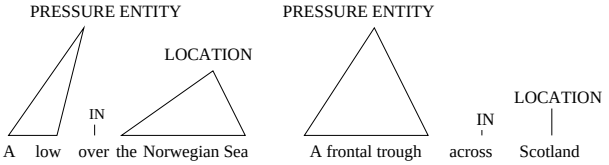


Figure 5: Abstracted tag sequences for two constituents

The parses with pruned trees from the preprocessing steps provide the necessary information for constituent matching. Figure 4 shows matching syntactic trees for two ARG0s from two propositions of a cluster. It is at this step that we use the domain entity tags to abstract away the constituents' syntactic tags. Figure 5 shows the constituents of Figure 4 with the tree structure reduced to tag sequences and domain entity types replacing the tags whenever necessary.

This abstraction step produces a number of unique domain entity augmented tag sequences for a particular semantic role. As a final step of template generation, we concatenate these abstracted constituent types for all the semantic roles in the given group.

To focus on template-like structures we only consider tag sequences that occur twice or more in the group.

The templates produced at the end of this step are essentially tag sequences interspersed with domain entities. In our definition of templates, the slots are the entity types and the fixed parts are constituted by word(s) used by the human experts for a particular tag sequence. Figure 6 shows some example templates. The upper case words in the figure correspond to the domain entities identified by the entity tagger and they form the slots in the templates. A total of 209 templates were produced.

PRESSURE_ENTITY expected over LOCATION by_0.5/on_0.5 DAY

PRESSURE_ENTITY to DIRECTION of LOCATION will drift slowly

WAVE will run_0.5/move_0.5 DIRECTION then DIRECTION

Associated PRESSURE_ENTITY will move DIRECTION across LOCATION TIME

Figure 6: Example Templates. Upper case tokens correspond to slots. For fixed parts, when there is a choice between words, the probability of the occurrence of words in that particular syntactic structure are tagged alongside.

4.2 A ROUGE Based Approach

ROUGE (Lin, 2004) is the standard automatic evaluation metric in the Summarization community. It is derived from the BLEU (Papineni et al., 2001) score which is the evaluation metric used in the Machine Translation community. The underlying idea in the metric is comparing the candidate and the reference sentences (or summaries) based on their token co-occurrence statistics. For example, a unigram based measure would compare the vocabulary overlap between the candidate and reference sentences. Thus, intuitively, we may use the ROUGE score as a measure for clustering the sentences. Amongst the various ROUGE statistics, the most appealing is Weighted Longest Common Subsequence(WLCS). WLCS favors contiguous LCS which corresponds to the intuition of finding the common template. We experimented with other ROUGE statistics but we got better and easily interpretable results using WLCS and so we chose it as the final metric. In all the approaches the data was first preprocessed (baseNP and NE tagged) as described in the previous subsection. In the following subsections, we describe the various clustering techniques that we tried using the ROUGE score followed by the alignment technique.

4.2.1 Clustering

Unsupervised Clustering: As the ROUGE score defines a distance metric, we can use this score for doing unsupervised clustering. We tried hierarchical clustering approaches but did not obtain good clusters, evaluated empirically. In empirical evaluation, we manually looked at the output clusters and made a judgement call whether the candidate clusters are reasonably coherent and potentially correspond to templates. The reason for the poor performance of the approach was the classical parameter estimation problem of determining a priori the number of clusters. We could not find an elegant solution for the problem without losing the motivation of an automated approach.

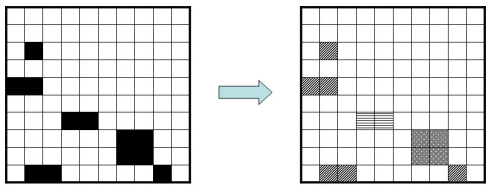


Figure 7: Deterministic clustering based on Graph connectivity. In the figure the squares with the same pattern belong to the same cluster.

Non-parametric Unsupervised Clustering: Since the unsupervised technique did not give good results, we experimented with a non-parametric clustering approach, namely, Cross-Association(Chakrabarti et al., 2004). It is a non-parametric unsupervised clustering algorithm for similarity (boolean) matrices. We obtain the similarity matrix in our domain by thresholding the ROUGE similarity score matrix. This technique also did not give us good clusters, evaluated empirically. The plausible reason for the poor performance seems to be that the technique is based on MDL (Minimum Description Length) principle. Since in our domain we expect a large number of clusters with small membership along many singletons, MDL principle is not likely to perform well.

Deterministic Clustering:

As the unsupervised techniques did not perform well, we tried deterministic clustering based on graph connectivity. The underlying intuition is that all the sentences $X_{1...n}$ that are "similar" to any other sentence Y_i should be in the same cluster even though X_j and X_k may not be "similar" to each other. Thus we find the connected components in the similarity matrix and label them as individual clusters.[4]

[4]This approach is similar to agglomerative single linkage clustering.

We created a similarity matrix by thresholding the ROUGE score. In the event, the clusters obtained by this approach were also not good, evaluated empirically. This led us to revisit the similarity function and tune it. We factored the ROUGE-WLCS score, which is an F-measure score, into its component Precision and Recall scores and experimented with various combinations of using the Precision and Recall scores. We finally chose a combined Precision and Recall measure (not f-measure) in which both the scores were independently thresholded. The motivation for the measure is that in our domain we desire to have high precision matches. Additionally we need to control the length of the sentences in the cluster for which we require a Recall threshold. F-measure (which is the harmonic mean of Precision and Recall) does not give us the required individual control. We set up our experiments such that while comparing two sentences the longer sentence is always treated as the reference and the shorter one as the candidate. This helps us in interpreting the Precision/Recall measures better and thresholding them accordingly. The approach gave us 149 clusters, which looked good on empirical evaluation. We can argue that using this modified similarity function for previous unsupervised approaches could have given better results, but we did not reevaluate those approaches as our aim of getting a reasonable clustering approach is fulfilled with this simple scheme and tuning the unsupervised approaches can be interesting future work.

4.3 Alignment

After obtaining the clusters using the Deterministic approach we needed to find out the template corresponding to each of the cluster. Fairly intuitively we computed the Longest Common Subsequence(LCS) between the sentences in each cluster which we then claim to be the template corresponding to the cluster. This resulted in a set of 149 templates, similar to the Predicate Argument based approach, as shown in figure 6.

5 Results

5.1 Evaluation Scheme

Since there is no standard way to evaluate template extraction for summary creation, we adopted a mix of subjective and automatic measures for evaluating the templates extracted. We define precision for this particular problem as:

$$precision = \frac{number\ of\ domain\ relevant\ templates}{total\ number\ of\ extracted\ templates}$$

This is a subjective measure and we undertook a study involving three subjects who were accustomed to the language used in the corpus. We asked the human subjects to mark each template as relevant or non-relevant to the weather forecast domain. We also asked them to mark the template as grammatical or ungrammatical if it is non-relevant.

Our other metric for evaluation is automatic recall. It is based on using the ROUGE-WLCS metric to determine a match between the preprocessed (baseNP and NE tagged) test corpora with the proposed set of correct templates, a set determined by taking an intersection of only the relevant templates marked by each judge. For the ROUGE based method, the test corpus consists of 262 sentences, while for the predicate-argument based method it consists of a set of 263 propositions extracted from the 262 sentences using ASSERT followed by a filtering of invalid propositions (e.g. ones starting with a verb). Amongst different ROUGE scores (precision/recall/f-measure), we consider precision as the criterion for deciding a match and experimented with different thresholding values.

Main Verb	Precision	Main Verb	Precision
deepen	0.67	weaken	0.83
expect	0.76	lie	0.57
drift	0.93	continue	0.97
build	0.95	fill	0.80
cross	0.78	move	0.86

Table 1: Precision for top 10 most frequently occurring verbs

5.2 Results: Predicate-Argument Based Approach

Table 1 shows the precision values for top 10 most frequently occurring verbs. (Since a major proportion ($> 90\%$) of the templates are covered by these verbs, we don't show all the precision values; it also helps to contain space.) The overall precision value achieved was 84.21%, the inter-rater Fleiss' kappa measure (Fleiss, 1971) between the judges being

$\kappa = 0.69$, demonstrating substantial agreement. The precision values are encouraging, and in most cases the reason for low precision is because of erroneous performance of the semantic role labeler system, which is corroborated by the percentage (47.47%) of ungrammatical templates among the irrelevant ones.

Results for the automated recall values are shown in Figure 8, where precision values are varied to observe the recall. For 0.9 precision in ROUGE-WLCS, the recall is 0.3 which shows that there is a 30% near exact coverage over propositions, while for 0.6 precision in ROUGE-WLCS, the recall is an encouraging 81%.

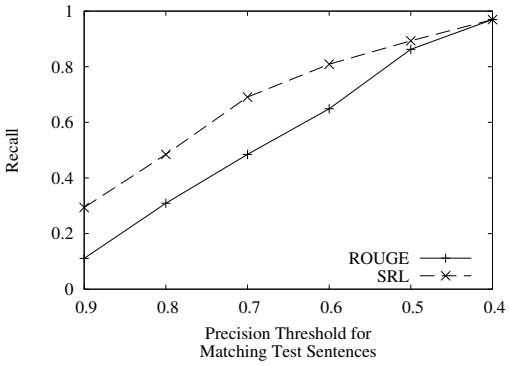


Figure 8: Automated Recall based on ROUGE-WLCS measure comparing the test corpora with the set of templates extracted by the Predicate-Argument (SRL) and the ROUGE based method.

5.3 Results: ROUGE based approach

Various precision and recall thresholds for ROUGE were considered for clustering. We empirically settled on a recall threshold of 0.8 since this produces the set of clusters with optimum number of sentences. The number of clusters and number of sentences in clusters at this recall values are shown in Figure 9 for various precision thresholds.

Precision was measured in the same way as the predicate argument approach and the value obtained was 76.3%, with Fleiss' kappa measure of $\kappa = 0.79$. The percentage of ungrammatical templates among the irrelevant ones was 96.7%, strongly indicating that post processing the templates using a parser can, in future, give substantial improvement. During error analysis, we observed simple grammatical errors in templates; first or last word being prepositions. So a fairly simple error recovery module that strips the leading and trailing prepositions was introduced. 20 templates out of the 149 were modified by the error recovery module and they were evaluated again by the three judges. The precision obtained for the modified templates was 35%, with Fleiss' kappa $\kappa = 1$, boosting the overall precision to 80.98%. The overall high precision is motivating as this is a fairly general approach that does not require any NLP resources. Figure 8 shows the automated recall values for the templates and abstracted sentences from the held-out dataset. For high precision points, the recall is low because there is not an exact match for most cases.

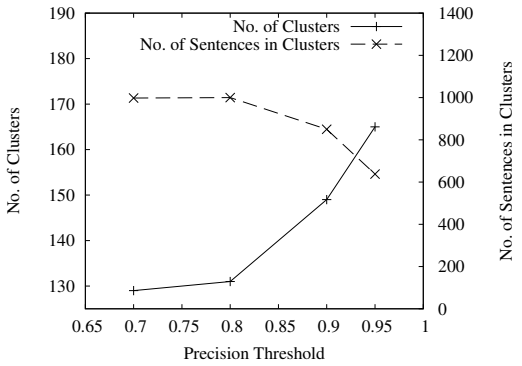


Figure 9: Number of clusters and total number of sentences in clusters for various Precision Thresholds at Recall Threshold=0.8

6 Conclusion and Future Work

In this paper, we described two new approaches for template extraction for briefing generation. For both approaches, high precision values indicate that meaningful templates are being extracted. However, the recall values were moderate and they hint at possible improvements. An interesting direction of future research is merging the two approaches and have one technique benefit from the other. The approaches seem complementary as the ROUGE based technique does not use the structure of the sentence at all whereas the predicate-argument approach is heavily dependent on it. Moreover, the predicate argument based approach gives general templates with one predicate while ROUGE based approach

can extract templates containing multiple verbs. It would also be desirable to establish the generality of the techniques, by using other domains such as newswire, medical reports and others.

Acknowledgements We would like to express our gratitude to William Cohen and Noah Smith for their valuable suggestions and inputs during the course of this work. We also thank the three anonymous reviewers for helpful suggestions. This work was supported by DARPA grant NBCHD030010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

ACE (2007). Automatic content extraction program. http://www.nist.gov/speech/tests/ace/.

Chakrabarti, D., Papadimitriou, S., Modha, D. S., & Faloutsos, C. (2004). Fully automatic cross-associations. *Proceedings of KDD '04* (pp. 79–88). New York, NY, USA: ACM Press.

Charniak, E. (2001). Immediate-head parsing for language models. *Proceedings of ACL '01* (pp. 116–123).

Collier, R. (1998). *Automatic template creation for information extraction*. Doctoral dissertation, University of Sheffield.

Elhadad, N., Kan, M.-Y., Klavans, J. L., & McKeown, K. (2005). Customization in a unified framework for summarizing medical literature. *Artificial Intelligence in Medicine*, *33*, 179–198.

Fellbaum, C. (1998). *WordNet – An Electronic Lexical Database*. MIT Press.

Filatova, E., Hatzivassiloglou, V., & McKeown, K. (2006). Automatic creation of domain templates. *Proceedings of COLING/ACL 2006* (pp. 207–214).

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin* (pp. 378–382).

Kingsbury, P., Palmer, M., & Marcus, M. (2002). Adding semantic annotation to the penn treebank. *Proceedings of the HLT'02*.

Kumar, M., Garera, N., & Rudnicky, A. I. (2007). Learning from the report-writing behavior of individuals. *IJCAI* (pp. 1641–1646).

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Proceedings of Workshop on Text Summarization*.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research Development*, *2*, 159–165.

Marsh, E., & Perzanowski, D. (1998). MUC-7 Evaluation of IE Technology: Overview of Results. *Proceedings of MUC-7*. Fairfax, Virginia.

Onyshkevych, B. (1994). Issues and methodology for template design for information extraction. *Proceedings of HLT '94* (pp. 171–176). Morristown, NJ, USA.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2001). Bleu: a method for automatic evaluation of machine translation.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., & Jurafsky, D. (2004). Shallow semantic parsing using support vector machines. *Proceedings of HLT/NAACL '04*. Boston, MA.

Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artif. Intell.*, *167*, 137–169.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. *AAAI/IAAI, Vol. 2* (pp. 1044–1049).

Surdeanu, M., Harabagiu, S., Williams, J., & Aarseth, P. (2003). Using predicate-argument structures for information extraction. *Proceedings of ACL 2003*.

Yangarber, R., Grishman, R., Tapanainen, P., & Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. *Proceedings of the 18th conference on Computational linguistics* (pp. 940–946). Morristown, NJ, USA.

Mining the Web for Relations between Digital Devices using a Probabilistic Maximum Margin Model

Oksana Yakhnenko
Iowa State University
Ames, IA, 50010
oksayakh@cs.iastate.edu

Barbara Rosario
Intel Research
Santa Clara, CA, 95054
barbara.rosario@intel.com

Abstract

Searching and reading the Web is one of the principal methods used to seek out information to resolve problems about technology in general and digital devices in particular. This paper addresses the problem of text mining in the digital devices domain. In particular, we address the task of detecting semantic relations between digital devices in the text of Web pages. We use a Naïve Bayes model trained to maximize the margin and compare its performance with several other comparable methods. We construct a novel dataset which consists of segments of text extracted from the Web, where each segment contains pairs of devices. We also propose a novel, inexpensive and very effective way of getting people to label text data using a Web service, the Mechanical Turk. Our results show that the maximum margin model consistently outperforms the other methods.

1 Introduction

In the digital home domain, home networks are moving beyond the common infrastructure of routers and wireless access points to include application-oriented devices like network attached storage, Internet telephones (VOIP), digital video recorders (e.g., Tivo), media players, entertainment PCs, home automation, and networked photo printers. There is an ongoing challenge associated with domestic network design, technology education, device setup, repair, and tuning. In this digital home setting, searching the Web is one of the principle methods used to seek out information and to resolve problems about technology in general and about digital devices in particular (Bly et al., 2006).

This paper addresses the problem of automatic text mining in the digital networks domain. Understanding the relations between entities in natural language sentences is a crucial step toward the goal of text mining. We address the task of identifying and extracting the sentences from Web pages which expressed a relation between two given digital devices in contrast to sentences in which these devices co-occur.

As an example, consider a user who is looking for information on digital video recorders (DVR), in particular, on how she can use a DVR with a PC. This user will not be satisfied with finding Web pages that simply mention these devices (such as the many products catalogs or shopping sites), but rather, the user is interested in retrieving and reading only the Web pages in which a specific relation between the two devices is expressed. The user is interested to learn that, for example, *"Any modern Windows PC can be used for DVR duty"* or that it is possible to transfer data from a DVR to a PC (*"You can simply take out the HD from the DVR, hook it up to the PC, and copy the videos over to the PC"*).[1]

The specific task addressed in this paper is the following: given a pair of devices, search the Web and extract only the sentences in which the devices are actually involved in an activity or a relation in the retrieved Web pages.

Note that we do not attempt to identify the type

[1] In italic are real sentences extracted from Web pages.

of relationship between devices but rather we classify sentences into whether the relation or activity is present or not, and thus we frame the problem as a binary text classification problem.[2] We propose a directed maximum margin probabilistic model to solve this classification task. Maximum margin probabilistic models have received a lot of attention in the machine learning and natural language processing literature. These models are trained to maximize the smallest difference between the probabilities of the true class and the best alternative class. Approaches such as maximum margin Markov networks (M3N) (Taskar et al., 2003) have been considered in prediction problems in which the goal is to assign a label to each word in the sentence or a document (such as part of speech tagging). It has also been shown that training of Bayesian networks by maximizing the margin can result in better performance than M3N in a flat-table structured domain (simulated and UCI repository datasets) and a structured prediction problem (protein secondary structure) (Guo et al., 2005). Given this background, we draw our attention to the application of maximum margin probabilistic models to a text classification task. We consider a *directed* model, where the parameters represent a probability distribution for words in each class (maximum margin equivalent of a Naïve Bayes). We evaluate the maximum margin model and compare its performance with the equivalent joint likelihood model (Naïve Bayes), conditional likelihood model (logistic regression) and support vector machines (SVM) on the relationship extraction task described above, as well as several other classification methods. Our results show that the maximum margin Naïve Bayes outperforms the other methods in terms of classification accuracy. To train such a model, manually labeled data is required, which is usually slow and expensive to acquire. To address this, we propose a novel, inexpensive and very effective way of getting people to label text data using the Mechanical Turk, an Amazon website[3] where people earn "micro-money" for completing tasks which are simple for humans to accomplish.

The paper is organized as follows: in Section 2 we discuss related work. In Section 3 we review joint likelihood and conditional likelihood models and maximum margin Naïve Bayes. In Section 4 we describe the collection of the training sentences, and how Mechanical Turk was used to construct the labels for the data. Section 5 introduces the experimental setup and presents performance results for each of the algorithms. We analyze Naïve Bayes, maximum margin Naïve Bayes and logistic regression in terms of the learned probability distributions in Section 6. Section 7 concludes with discussion.

2 Related work

2.1 Relation extraction

There has been a spate of work on relation extraction in recent years. However, many papers actually address the task of role extraction: (usually two) entities are identified and the relationship is *implied* by the co-occurrence of these entities or by some linguistic expression (Agichtein and Gravano, 2000; Zelenko et al., 2003).

Several papers propose the use of machine learning models and probabilistic models for relation extraction: Naïve Bayes for the relation *subcellular-location* in the bio-medical domain (Craven, 1999) or for *person-affiliation* and *organization-location* (Zelenko et al., 2003). Rosario and Hearst (2005) have used a more complicated dynamic graphical model to identify interaction types between proteins and to simultaneously extract the proteins.

2.2 Maximum margin models

Probabilistic graphical models and different approaches to training them have received a lot of attention in application to natural language processing. McCallum and Nigam (1998) showed that Naïve Bayes can be a very accurate model for text categorization.

Since probabilistic graphical models represent joint probability distributions whereas classification focuses on the conditional probability, there has been debate regarding the objective that should be maximized in order to train these models. Ng and Jordan (2001) have compared a joint likelihood

[2] Classifying or clustering the relation types would involve the tricky task of defining the possible semantic relations between devices as well as relations. We plan of addressing this in the future work, however, we believe that such binary distinction is already quite useful for many tasks in this domain.

[3] Available at http://www.mturk.com

model (Naïve Bayes) and its discriminative counterpart (logistic regression), and they have shown that while for large number of examples logistic regression has a lower error rate, Naïve Bayes often outperforms logistic regression for smaller data sets. However, Klein and Manning (2002) showed that for natural language and text processing tasks, conditional models are usually better than joint likelihood models. Yakhnenko et al. (2005) also showed that conditional models suffer from overfitting in text and sequence structured domains.

In recent years, the interest in learning parameters of probabilistic models by maximizing the probabilistic margin has developed. Taskar et al. (2003) have solved the problem of learning Markov networks (undirected graphs) by maximizing the margin. Their work has focused on likelihood based structured classification where the goal is to assign a class to each word in the sentence or a document. Guo et al. (2005) have proposed a solution to learning parameters of the maximum margin Bayesian Networks.

Surprisingly, little has been done in applying probabilistic models trained to maximize the margin to simple classification tasks (to the best of our knowledge). Therefore, since the Naïve Bayes model has been shown to be a successful algorithm for many text classification tasks (McCallum and Nigam, 1998) we suggest learning the parameters of Naïve Bayes model to maximize the probabilistic margin. We apply the Naïve Bayes model trained to maximize the margin to a relation extraction task.

3 Joint and conditional likelihood models and maximum margin

We now describe the background in probabilistic models as well as different approaches to parameter estimation for probabilistic models. In particular, we describe Naïve Bayes, logistic regression (analogous to conditionally trained Naïve Bayes) and then introduce Naïve Bayes trained to maximize the margin.

First, we introduce some notation. Let D be a corpus that consists of training examples. Let T be the size of D. We represent each example with a tuple $\langle s, c\rangle$ where s is a sentence or a document, and c is a label from a set of all possible labels, $c \in C = \{c_1 ... c_m\}$. Let $D=\{\langle s^i, c^i\rangle\}$ where superscript $1 \leq i \leq T$ is the index of the document in the corpus, and c^i is the label of example s^i. Let V be vocabulary of D, so that every document s consists of elements of V. We will use s_j to denote a word from s in position j, where $1 \leq j \leq length(s)$.

3.1 Generative and discriminative Naïve Bayes models

A probabilistic model assigns to each instance s a joint probability of the instance and the class $P(s, c)$. If the probability distribution is known, then a new instance s_{new} can be classified by giving it a label which has the highest probability:

$$c = \arg\max_{c_k \in C} P(c_k | s_{new}) \quad (1)$$

Joint likelihood models learn the parameters by maximizing the probability of an example and its class, $P(s, c)$. Naïve Bayes multinomial, for instance, assumes that all words in the sentence are independent given the class, and computes this probability as $P(c)\prod_{j=1}^{length(s)} P(s_j|c)$. Each of $P(s_j|c)$ and $P(c)$ are estimated from the training data using relative frequency estimates. From here on we will refer to joint likelihood Naïve Bayes multinomial as NB-JL.

Since the conditional probability is needed for the classification task, it has been suggested to solve the maximization problem and train the model so that the choice of the parameters maximizes $P(c|s)$ directly. One can use a joint likelihood model to obtain joint probability distribution $P(s, c)$ and then use the definition of conditional probability to get $P(c|s) = P(s, c)/\sum_{c_k \in C} P(s, c_k)$. The solutions that maximize this objective function are searched for by using gradient ascent methods. Logistic regression is a conditional model that assumes the independence of features given the class, and it is a conditional counterpart to NB-JL (Ng and Jordan, 2001).

We will now introduce a probabilistic maximum margin objective and describe a maximum margin model that is analogous to Naïve Bayes and logistic regression.

3.2 Maximum margin training of Naïve Bayes models

The basic idea behind maximum margin models is to choose model parameters that for each example will make the probability of the true class and the example as high as possible while making the probability of the nearest alternative class as low as possible. Formally, the maximum margin objective is

$$\gamma = \min_{i=1}^{T} \min_{c \neq c^i} \frac{P(c^i|s^i)}{P(c|s^i)} = \min_{i=1}^{T} \min_{c \neq c^i} \frac{P(s^i, c^i)}{P(s^i, c)} \quad (2)$$

Here $P(s, c)$ is modeled by a generative model, and parameter learning is reduced to solving a convex optimization problem (Guo et al., 2005).

In order for the example to be classified correctly, the probability of the true class given the example has to be higher than the probability of getting the wrong class or

$$\gamma_i = \log p(c^i|s^i) - \log p(c^j|s^i) > 0 \quad (3)$$

where $j \neq i$ and c^i is the true label of example s^i. The larger the margin γ_i is, the more confidence we have in the prediction.

We consider a Naïve Bayes model trained to maximize the margin and refer to this model as MMNB. Using exponential family notation, let $P(s_j|c) = e^{\mathbf{w}_{s_j|c}}$. The likelihood is $P(s, c) = e^{\mathbf{w}_c} \prod_{j=1}^{len(s)} e^{\mathbf{w}_{s_j|c}}$. Then the log-likelihood

$$\log P(s, c) = \mathbf{w_c} + \sum_{j=1}^{len(s)} count(s_j)\mathbf{w}_{s_j|c} = \mathbf{w} \cdot \phi(s, c) \quad (4)$$

where $\mathbf{w}$ is the weight vector for all the parameters that need to be learned, and $\phi(s, c)$ is the vector of counts of words associated with each parameter $\phi(s, c) = (...count(s_j c)....)$ in s for class c.

The general formulation for Bayesian networks was given in Guo et al., and we adapt their formulation for training a Naïve Bayes model. The parameters are learned by solving a convex optimization problem. If the margin γ is the smallest log-ratio, then γ needs to be maximized, where the constraint is that for each instance the log-ratio of the probability of predicting the instance correctly and predicting it incorrectly is at least γ. Such formulation also allows for the use of slack variables ξ so that the classifier "gives up" on the examples that are difficult to classify.

$$\begin{aligned}
&\text{minimize}_{\ \gamma, \mathbf{w}, \xi} \ \frac{1}{\gamma^2} + B \sum_{i=1}^{T} \xi_i \\
&\text{subject to } \mathbf{w}(\phi(i, c^i) - \phi(i, c)) \geq \gamma \delta(c^i, c) - \xi_i \\
&\text{and } \sum_{s_i \in V} e^{\mathbf{w}_{s_i, c}} \leq 1 \forall c \in C \\
&\text{and } \gamma \geq 0
\end{aligned}$$

This problem is convex in the variables $\gamma, \mathbf{w}, \epsilon$. B is a regularization parameter, and $\delta(c^i, c) = 1$ if $c^i \neq c$ and 0 otherwise. The inequality constraint for probabilities is needed to preserve convexity of the problem, and in the case of Naïve Bayes, the probability distribution over the parameters (the equality constraint) can be easily obtained by renormalizing the learned parameters.

The minimization problem is somewhat similar to ℓ_2-norm support vector machine with a soft margin (Cristianini and Shawe-Taylor, 2000). The first constraint imposes that for each example the log of the ratio between the example under the true class and the example under some alternative class is greater than the margin allowing for some slack. The second constraint enforces that the parameters do not get very large and that the probabilities sum to less than 1 to maintain valid probability distribution (the inequality constraint is required to preserve convexity, and the probability distribution can be obtained after training by renormalization).

Following Guo et al. (2005), we find parameters using a log-barrier method (Boyd and Vandenberghe, 2004), the sum of the logarithms of constraints are subtracted from the objective and scaled by a parameter μ. The problem is solved sequentially using a fixed μ and gradually lowering μ to 0. The solution for a fixed μ is obtained using (typically) a second order method to guarantee faster convergence. This solution is then used as the initial parameter values for the next μ. In our implementation we used a limited memory quasi-Newton method (Nocedal and Liu, 1989).

4 Data and labels

4.1 The problem of labeling data

One major problem of natural language processing is the sparsity of data; to accurately learn a linguistic model, one needs to label a large amount of text, which is usually an expensive requirement. For information extraction, the labeling process is particularly difficult and time consuming. Moreover, in different applications one needs different labeled data for each domain. We propose a creative way of convincing many people to label data quickly and at low cost to us by using the Mechanical Turk. Similarly, Luis von Ahn (2006) creates very successful and compelling computer games in such a way that while playing, people provide labels for images on the Web.

4.2 Collecting data and label agreement analysis

To collect the data, we identified 58 pairs of digital devices, as well as their synonyms (for example, computer, laptop, PC, desktop, etc), and different manufacturers for a given device (for example Toshiba, Dell, IBM, etc). The devices alone were used to construct the query (for example 'computer, camera', as well as a combination of manufacturer and devices (for example 'dell laptop, cannon camera'). Each of these pairs was used as a query in Google, and the sentences that contain both devices were extracted resulting in a total of 3624 sentences. We use the word 'sentence' when referring to the examples, however we note that not all text excerpts are sentences, some are chunks of text data.

To label the data we used the Mechanical Turk (MTurk), a Web service that allows you to create and post a task for humans to solve; typical tasks are labeling pictures, choosing the best among several photographs, writing product descriptions, proofreading and transcribing podcasts. After the task is completed the requesters can then review the submissions and reject them if the results are poor.

We created a total of 121 unique surveys consisting of 30 questions. Each question consisted of one of the extracted statements with the devices highlighted in red. The task for the labeler was to choose between 'Yes', if the statement contained a relation between the devices, 'No' if it did not, or 'not applicable' if the text extract was not a sentence, or if the query words were not used as different devices (as for noun compounds such as *computer stereo*).[4] Each survey was assigned to 3 distinct workers, thus having 3 possible labels for all 3624 sentences.[5]

		worker3		
worker1	worker2	yes	no	n/a
yes	yes	**1091**	237	23
	no	226	281	22
	n/a	19	18	6
no	yes	217	199	8
	no	186	**870**	56
	n/a	14	39	8
n/a	yes	17	13	5
	no	6	32	6
	n/a	4	12	**9**

Table 1: Summary of the labels assigned by the MT workers to all the sentences.

We used Fleiss's kappa (Fleiss, 1971) (a generalization of kappa statistic which takes into account multiple raters and measures inter-rater reliability) in order to determine the degree of agreement and to determine whether the agreement was accidental. Kappa statistics is a number between 0 and 1 where 0 is random agreement, and 1 is perfect agreement.

In order to compute kappa statistic, since the computation requires that the raters are the same for each survey, we mapped workers into 'worker1', 'worker2', 'worker3' with 'worker1' being the first worker to complete each of the 121 surveys, 'worker2' the second, and so on. The responses are summarized in Table 1.

The overall Fleiss's kappa was 0.41[6], and therefore, it can be concluded that the agreement between the workers was not accidental.

We had perfect agreement for 49% of all sentences, 5% received all three labels (these examples were discarded) and for the remaining 46% two la-

[4]This dataset, including all the MTurk's workers responses is available at http://www.cs.iastate.edu/~oksayakh/relation_data.html

[5]The requirement for the workers to be different was imposed by the MTurk system, which checks their Amazon identity; however, this still allows for the same person who has multiple identities to complete the same task more than once.

[6]The kappa coefficients for categories 'Yes' and 'No' were 0.45 and 0.41 respectively (moderate agreement) and for category 'not applicable' was 0.15 (slight agreement).

bels were assigned (the majority vote was used to determine the final label). For these cases, we noticed that some of the labels were wrong (however in most cases the majority vote results in the correct label) but other sentences were ambiguous and either label could be right. To assign the final label we used majority vote, and we discarded sentences for which 'not applicable' was the majority label.

We rewarded the users with between 15 and 30 cents per survey (resulting in less than a cent for a text segment) and we were able to obtain labels for 3594 text segments for under $70. It also took anywhere between a few minutes to a half-hour from the time the survey was made available until it was completed by all three users. We find Mechanical Turk to be a quite interesting, inexpensive, fairly accurate and fast way to obtain labeled data for natural language processing tasks.

We used this data to evaluate the classification models as described in the next section.

5 Experimental setup and results

The words were stemmed, and the data was smoothed by mapping all the words that appeared only once to a unique token `smoothing_token` (resulting in a total of approximately 2,800 words in the vocabulary). We performed 10-fold cross-validation, with smoothed test data where all the unseen words in the test data were mapped to the token `smoothing_token`. We used the exact same data in the folds for all four algorithms – MMNB, NB-JL, logistic regression and SVM. Since MMNB, SVM, and logistic regression allows for regularization, we used tuning to find the optimal performance of the models. At each fold we withheld 30% of the training data for validation purposes (thus resulting in 3 disjoint sets at each fold). The model was trained on the resulting 70% of the training data for different values of the regularization parameters, and the value which yielded the highest accuracy on the validation set was used to train the model that was evaluated on the test set.

As a baseline, we consider a classifier which assigns the most frequent label ('Yes'); such a classifier results in 53% accuracy.

Table 2 summarizes the performance of MMNB and other algorithms as determined by 10-fold cross-validation with tuning data. We compared the accuracies of the maximum margin model with the accuracy of generative Naïve Bayes, logistic regression and SVM as shown in Table 2. The MMNB has the highest accuracy followed by NB-JL and then SVM with RBF kernel. Even after tuning, logistic regression did not reach the performance of MMNB and NB-JL.

Algorithm	Accuracy
MMNB	**80.23%**
SVM-RBF	76.49%
NB-JL	75.62%
Perceptron	74.04%
SVM-2	72.72%
SVM-3	71.54%
DT	70.76%
LR	69.95%
SVM-1	69.94%
Baseline	53.8%

Table 2: Classification accuracies as determined by 10-fold cross-validation. SVM-1 uses linear kernel, SVM-2 uses quadratic kernel, SVM-3 uses cubic kernel, SVM-RBF uses RBF kernel with parameter $\gamma = 0.1$. The Decision Tree (DT) uses binary splits. LR is logistic regression.

Since MMNB is trained to maximize the margin, we compared it with the Support Vector Machine (linear maximum margin classifier). Counts of words were used as features (resulting in the bag of words representation[7]). We ran our experiments with linear, quadratic, cubic and RBF kernels. SVM was tuned using the validation set similarly to MMNB. We also experimented with Perceptron and Decision Tree using binary splits with reduced error-pruning, which are methods commonly used for text classification (due to lack of space, we will not describe these methods and their applications, but refer the reader to Manning and Schütze (1999)). Among all the known methods, the maximum margin Naïve Bayes is the algorithm with the highest accuracy, suggesting that it is a competitive algorithm in relation extraction and text classification tasks.

[7] This representation allows for additional or alternative features such as k-grams of words, whether the words are capitalized, where on the page the sentence was located, etc. Evaluating MMNB and other methods with additional features is of interest in the future

6 Analysis of behavior of Naïve Bayes, maximum margin Naïve Bayes and logistic regression

We analyzed the behavior of the parameters of the probabilistic models (Naïve Bayes, MMNB and logistic regression) on the training data. For each example in the training data we computed the probability $P(c = noRelation|s)$ using the parameters from the model, and examined the probabilities assigned to examples from both classes. We show these plots in Figure 1.

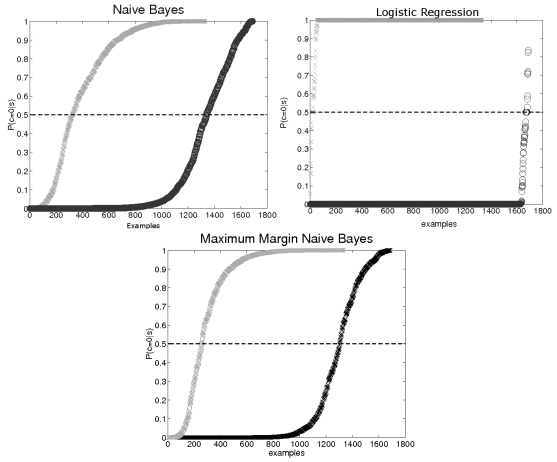


Figure 1: Probability distribution of $P(c = noRelation|s)$ learned by the Naïve Bayes (upper left), logistic regression (upper right) and maximum margin Naïve Bayes(lower). In gray are class-conditional probabilities assigned to positive examples, and in black are class-conditional probabilities assigned to negative examples.

As we see, the logistic regression discriminates between the majority of the examples by assigning extreme probabilities (0 and 1). However, there are some examples which are extremely borderline, and thus it does not generalize well on the test set. On the other had, Naïve Bayes does not have such "sharp" discrimination. Maximum margin Naïve Bayes has "sharper" discrimination than Naïve Bayes, however the discrimination is smoother than for logistic regression. The examples which are more difficult to classify have probabilities that are more spread out (away from 0.5), as opposed to the case of logistic regression, which assigns these difficult examples to probability close to 0.5. This suggests that maximum margin Naïve Bayes, possibly has a better generalization ability than both logistic regression and Naïve Bayes, however to make such a claim additional experiments are needed.

7 Conclusions

The contribution of this paper is threefold. First, we addressed the important problem of identifying the presence of semantic relations between entities in text, focusing on the digital domain. We presented some encouraging results; it remains to be seen however, how this would transfer to better results in an information retrieval task. Secondly, we considered a probabilistic model trained to maximize the margin, that achieved the highest accuracy for this task, suggesting that it could be a competitive algorithm for relation extraction and text classification in general. However in order to fully evaluate the MMNB method for relation classification it needs to be applied to other classification and or relation prediction tasks. We also empirically analyzed the behavior of the parameters learned by maximum margin model and showed that the parameters allow for better generalization power than Naïve Bayes or logistic regression models. Finally, we suggested an inexpensive way of getting people to label text data via Mechanical Turk.

Acknowledgment The authors would like to thank the reviewers for their feedback and comments; William Schilit for invaluable insight and help and for first suggesting using the MTurk to gather labeled data; David McDonald for help with developing survey instructions; and numerous MT workers for providing the labels.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of Digital Libraries*.

Sara Bly, William Schilit, David McDonald, Barbara Rosario, and Ylian Saint-Hilaire. 2006. Broken expectations in the digital home. In *Proceedings of Computer Human Interaction (CHI)*.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Mark Craven. 1999. Learning to extract relations from Medline. In *AAAI-99 Workshop*.

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Yuhong Guo, Dana Wilkinson, and Dale Schuurmans. 2005. Maximum margin bayesian networks. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, page 233.

Dan Klein and Christopher Manning. 2002. Conditional structure versus conditional estimation in nlp models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, June.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*.

Andrew Y. Ng and Michael I. Jordan. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 841–848.

Jorge Nocedal and Dong C. Liu. 1989. On the limited memory method for large scale optimization. *Mathematical Programming*, 3(45):503–528.

Barbara Rosario and Marti Hearst. 2005. Multi-way relation classification: Application to protein-protein interactions. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Benjamin Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *Proceedings of Neural Information Processing Systems (NIPS)*.

Luis von Ahn. 2006. Games with a purpose. *Computer*, 39(6):92–94.

Oksana Yakhnenko, Adrian Silvescu, and Vasant Honavar. 2005. Discriminatively trained markov model for sequence classification. In *Proceedings of International Conference on Data Mining (ICDM)*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Learning Patterns from the Web to Translate Named Entities for Cross Language Information Retrieval

Yu-Chun Wang[†‡] **Richard Tzong-Han Tsai**[§*] **Wen-Lian Hsu**[†]

[†]Institute of Information Science, Academia Sinica, Taiwan
[‡]Department of Eletrical Engineering, National Taiwan University, Taiwan
[§]Department of Computer Science and Engineering, Yuan Ze University, Taiwan
albyu@iis.sinica.edu.tw
thtsai@saturn.yzu.edu.tw
hsu@iis.sinica.edu.tw
[*]corresponding author

Abstract

Named entity (NE) translation plays an important role in many applications. In this paper, we focus on translating NEs from Korean to Chinese to improve Korean-Chinese cross-language information retrieval (KCIR). The ideographic nature of Chinese makes NE translation difficult because one syllable may map to several Chinese characters. We propose a hybrid NE translation system. First, we integrate two online databases to extend the coverage of our bilingual dictionaries. We use Wikipedia as a translation tool based on the inter-language links between the Korean edition and the Chinese or English editions. We also use Naver.com's people search engine to find a query name's Chinese or English translation. The second component is able to learn Korean-Chinese (K-C), Korean-English (K-E), and English-Chinese (E-C) translation patterns from the web. These patterns can be used to extract K-C, K-E and E-C pairs from Google snippets. We found KCIR performance using this hybrid configuration over five times better than that a dictionary-based configuration using only Naver people search. Mean average precision was as high as 0.3385 and recall reached 0.7578. Our method can handle Chinese, Japanese, Korean, and non-CJK NE translation and improve performance of KCIR substantially.

1 Introduction

Named entity (NE) translation plays an important role in machine translation, information retrieval, and question answering. It is a challenging task because, although there are many online bilingual dictionaries, they usually lack domain specific words or NEs. Furthermore, new NEs are generated everyday, but bilingual dictionaries cannot update their contents frequently. Therefore, it is necessary to construct a named entity translation (NET) system.

Economic ties between China and Korea have become closer as China has opened its markets further, and demand for the latest news and information from China continues to grow rapidly in Korea. One key way to meet this demand is to retrieve information written in Chinese by using Korean queries, referred to as Korean-Chinese cross-language information retrieval (KCIR). The main challenge involves translating NEs because they are usually the main concepts of queries. In (Chen et al., 1998), the authors romanized Chinese NEs and selected their English transliterations from English NEs extracted from the Web by comparing their phonetic similarities with Chinese NEs. Yaser Al-Onaizan (Al-Onaizan and Knight, 2002)

transliterated an NE in Arabic into several candidates in English and ranked the candidates by comparing their counts in several English corpora. Unlike the above works, whose target languages are alphabetic, in K-C translation, the target language is Chinese, which uses an ideographic writing system. Korean-Chinese NET is much more difficult than NET considered in previous works because, in Chinese, one syllable may map to tens or hundreds of characters. For example, if an NE written in Korean comprises three syllables, there may be thousands of possible translation candidates in Chinese.

In this paper, we propose an effective hybrid NET method which can help improve performance of cross-language information retrieval systems. We also describe the construction of a Korean-Chinese CLIR system able to evaluate the effectiveness of our NE translation method.

2 Difficulties in Korean-Chinese Named Entity Translation for IR

2.1 Korean NET

Most Korean NEs originate from Hanja. Therefore, the most straightforward way to translate a Korean name into Chinese is to use its Hanja equivalent. Take the name of Korea's president, "노무현" (No Mu-hyeon), as an example. We can directly convert it to its Hanja equivalent: "盧武鉉" (Lu Wu-Xuan). Or in the case of the city name "부산" (Pusan/釜山/Fu-shan) and the company name "삼성" (Samsung/三星/San-xing), Chinese also presents Hanja equivalents.

If the Hanja name is unknown, the name is translated character by character. Each Hangul character is basically translated into a corresponding Hanja character. For example, the name of the Korean actor "조인성" (Cho In-seong) is usually translated as "趙仁成" (Zhao Ren-cheng) because '조' is mapped to '趙', '인' mapped to '仁', and '성' mapped to '成'. However, that translation may differ from the person's given Hanja name.

For native Korean NEs which have no corresponding Hanja characters, we must turn to transliteration or convention. Take the name of South Korea's capital "서울" (Seoul) as an example. Before 2005, Chinese media and government used the old Hanja name of the city "漢城" (Han-cheng), which was used during Joseon dynasty (A.D. 1392–1910). However, after 2005, Chinese switched to using the transliteration "首爾" (Shou-er) instead of "漢城" at the request of the Seoul Metropolitan Government. This example illustrate how more than one Chinese translation for a Korean name is possible, a phenomenon which, at times, makes Korean-Chinese information retrieval more difficult.

2.2 Chinese NET

To translate a Chinese NE written in Hangul, we begin by considering the two C-K NET approaches. The older is based on the Sino-Korean pronunciation and the newer on the Mandarin.

For example, "臺灣" (Taiwan) used to be transliterated solely as "대만" (Dae-man). However, during the 1990s, transliteration based on Mandarin pronunciation became more popular. Presently, the most common transliteration for "臺灣" is "타이완" (Ta-i-wan), though the Sino-Korean-based "대만" is still widely used. For Chinese personal names, both ways are used. For example, the name of Chinese actor Jackie Chan ("成龍" Cheng-long) is variously transliterated as "성룡" Seong-ryong (Sino-Korean) and "청룽" Cheong-rung (Mandarin).

Translating Chinese NEs by either method is a major challenge because each Hangul character may correspond to several different Chinese characters that have similar pronunciations in Korean. This results in thousands of possible combinations of Chinese characters, making it very difficult to choose the most widely used one one.

2.3 Japanese NET

Japanese NEs may contain Hiraganas, Katakanas, or Kanjis. For each character type, J-C translation rules may be similar to or very different from K-C translation rules. Some of these rules are based on Japanese pronunciation, while some are not. For NEs composed of all Kanjis, their Chinese translations are generally exactly the same as their Kanji written forms. In contrast, Japanese NEs

are transliterated into Hangul characters. Take "名古屋" (Nagoya) for example. Its Chinese translation "名古屋" is exactly the same as its Kanji written form, while its pronunciation (Ming Gu Wu) is very different from its Japanese pronunciation. This is different from its Korean translation, "나고야" (Na go ya). In this example, we can see that, because the translation rules in Chinese and Korean are different, it is ineffective to utilize phonetic similarity to find the Chinese translation equivalent to the Korean translation.

2.4 Non-CJK NET

In both Korean and Chinese, transliteration methods are mostly used to translate non-CJK NEs. Korean uses the Hangul alphabet for transliteration. Because of the phonology of Korean, some phonemes are changed during translation because the language lacks these phonemes. (Oh, 2003; Lee, 2003) In contrast, Chinese transliterates each syllable in a NE into Chinese characters with similar pronunciation. Although there are some conventions for selecting the transliteration characters, there are still many possible transliterations since so many Chinese characters have the same pronunciation. For instance, the name "Greenspan" has several Chinese transliterations, such as "葛林斯班" (Ge-lin-si-ban) and "葛林斯潘" (Ge-lin-si-pan). In summary, it is difficult to match a non-CJK NE transliterated from Korean with its Chinese transliteration due to the latter's variations.

3 Our Method

In this section, we describe our Korean-Chinese NE translation method for dealing with the problems described in Section 2. We either translate NE candidates from Korean into Chinese directly, or translate them into English first and then into Chinese. Our method is a hybrid of two components: extended bilingual dictionaries and web-based NET.

3.1 Named Entity Candidate Selection

The first step is to identify which words in a query are NEs. In general, Korean queries are composed of several eojeols, each of which is composed of a noun followed by the noun's postposition, or a verb stem followed by the verb's ending. We remove the postposition or the ending to extract the key terms, and then select person name candidates from the key terms. Next, the maximum matching algorithm is applied to further segment each term into words in the Daum Korean-Chinese bilingual dictionary[1]. If the length of any token segmented from a term is 1, the term is regarded as an NE to be translated.

3.2 Extension of Bilingual Dictionaries

Most NEs are not included in general bilingual dictionaries. We adopt two online databases to translate NEs: Wikipedia and Naver people search.

3.2.1 Wikipedia

In Wikipedia, each article has an interlanguage link to other language editions, which we exploit to translate NEs. Each NE candidate is first sent to the Korean Wikipedia, and the title of the matched article's Chinese version is treated as the NE's translation in Chinese. However, if the article lacks a Chinese version, we use the English edition to acquire the NE's translation in English. The English translation is then transliterated into Chinese by the method described in Section 3.3.3.

3.2.2 Naver People Search Engine

Most NEs are person names that cannot all be covered by the encyclopedia. We use Naver people search engine to extend the coverage of person names. Naver people search is a translation tool that maintains a database of famous people's basic profiles. If the person is from CJK, the search engine returns his/her name in Chinese; otherwise, it returns the name in English. In the former case, we can adopt the returned name directly, but in the latter, we need to translate the name into Chinese. The translation method is described in Section 3.3.3.

[1] http://cndic.daum.net

3.3 Translation Pattern from the Web

Obviously, the above methods cannot cover all possible translations of NEs. Therefore, we propose a pattern-based method to find the translation from the Web. Since the Chinese translations of some NEs cannot be found by patterns, we find their Chinese translations indirectly by first finding their English translations and then finding the Chinese translations. Therefore, we must generate K-C patterns to extract K-C translation pairs, as well as K-E and E-C patterns to extract K-E and E-C pairs, respectively.

3.3.1 Translation Pattern Learning

Our motivation is to learn patterns for extracting NEs written in the source language and their equivalents in the target language from the Web. First, we need to prepare the training set. To generate K-C and K-E patterns, we collect thousands of NEs that originated in Korean, Chinese, Japanese, or non-CJK languages from Dong-A Ilbo (a South Korean newspaper). Then, all the Korean NEs are translated into Chinese manually. NEs from non-CJK languages are also translated into English. To generate E-C patterns, we collect English NEs from the MUC-6 and MUC-7 datasets and translate them into Chinese manually.

We submit each NE in the source language (source NE) and its translation in the target language as a query to Google search engine. For instance, the Korean NE "메이저리그" and its translation "Major League" are first composed as a query "+메 이 저 리 그 + Major League", which is then sent to Google. The search engine will return the relevant web documents with their snippets. We collect the snippets in the top 20 pages and we break them into sentences. Only the sentences that contain at least one source NE and its translation are retained.

For each pair of retained sentences, we apply the Smith-Waterman local alignment algorithm to find the longest common string, which is then added to the candidate pattern pool. During the alignment process, positions where the two input sequences share the same word are counted as a match. The following is an example of a pair of sentences that contains "메이저리그" and its English translation, "Major League":

- "**메이저리그**(**Major League**)는 수많은 산고 끝에 탄생한 산물입니다"
- "미국 **메이저리그**(**Major League**)는,"

After alignment, the pattern is generated as:

<*Korean NE*>(<*English Translation*>)는

This pattern generation process is repeated for each NE-translation pair.

3.3.2 Translation Pattern Filtering

After learning the patterns, we have to filter out some ineffective patterns. First, we send a Korean NE, such as "메 이 저 리 그", to retrieve the snippets in the top 50 pages. Then, we apply all the patterns to extract the translations from the snippets. The correct rate of each translation pattern is calculated as follows: $CorrectRate = C_{correct}/C_{all}$, where $C_{correct}$ is the total number of correct translations extracted by the pattern and C_{all} is the total number of translations extracted by the pattern. If the correct rate of the pattern is below the threshold τ, the pattern will be dropped.

3.3.3 Pattern-Based NET

The translations of some NEs, especially from CJK, can be found comparatively easily from the Web. However, for other NEs, especially from non-CJK, this is not the case. Therefore, we split the translation process into two stages: the first translates the NE into its English equivalent, and the second translates the English equivalent into Chinese.

To find an NE's Chinese translation, we first apply the translation patterns to extract possible Chinese translations. If its Chinese translation cannot be found, the K-E patterns are used to find its English translation instead. If its English translation can be found, the E-C patterns are then used to find its Chinese translation.

4 System Description

We construct a Korean-Chinese cross language information retrieval (KCIR) system to determine how our person name translation methods affect KCIR's performance. A Korean query is

translated into Chinese and then used to retrieve Chinese documents. The following sections describe the four stages of our KCIR system. We use an example query, "코스보의 사태, 나토, 유엔" (Kosovo's situation, NATO, UN), to demonstrate the work flow of our system.

4.1 Query Processing

Unlike English, Korean written texts do not have word delimiters. Spaces in Korean sentences separate eojeols. First, the postposition or verb ending in each eojeol is removed. In our example query, we remove the possessive postposition "의" at the end of the first eojeol. Then, NE candidates are selected using the method described in Section 3.1. "코스보" (Kosovo) is recognized as an NE, and other terms "사태" (situation), "나토" (NATO), and "유엔" (UN) are general terms because they can be found in the bilingual dictionary.

4.1.1 Query Translation

Terms not selected as NE candidates are sent to the online Daum Korean-Chinese dictionary and Naver Korean-Chinese dictionary[2] to get their Chinese translations. In our example, the terms "사태" (situation), "나토" (NATO), and "유엔" (UN) can be correctly translated into Chinese by the bilingual dictionaries as "事態" (situation), "北大西洋公約組織" (NATO), and "聯合國" (UN), respectively.

We employ Wikipedia, Naver people search, and the pattern-based method simultaneously to translate the NE candidate "코스보" (Kosovo). Up to now, there is no article about Kosovo in Korean Wikipedia. Naver people search does not contain an article either because it is not a person name. Meanwhile, since the K-C translation patterns cannot extract any Chinese translations, the K-E patterns are used to get the English translations, such as "Kosovo", "Cosbo", and "Kosobo". The E-C patterns are then employed to get the Chinese translation from the three English translations. Among them, only Chinese translations for "Kosovo" can be found because the other two are either wrong or rarely used translations. The Chinese translations extracted by our patterns are "科索夫" (Ke-suo-fu), "科索伏" (Ke-suo-fu), and "科索沃" (Ke-suo-wuo). They are all correct transliterations.

[2] http://cndic.naver.com

4.2 Term Disambiguation

A Hangul word might have many meanings. Besides, sometimes the translation patterns might extract wrong translations of the NE. This phenomenon causes ambiguities during information retrieval and influence the performance of IR significantly. To solve this problem, we adopt the mutual information score (MI score) to evaluate the co-relation between a translation candidate tc_{ij} for a term qt_i and all translation candidates for all the other terms in Q; tc_{ij}'s MI score given Q is calculated as follows:

$$\text{MI score}(tc_{ij}|Q) = \sum_{x=1,x\neq i}^{|Q|} \sum_{y=1}^{Z(qt_x)} \frac{Pr(tc_{ij}, tc_{xy})}{Pr(tc_{ij})Pr(tc_{xy})}$$

where $Z(qt_x)$ is the number of translation candidates of the x-th query term qt_x; tc_{xy} is y-th translation candidate for qt_x; $\Pr(tc_{ij}, tc_{xy})$ is the probability that tc_{ij} and tc_{xy} co-occur in the same sentence; and $\Pr(tc_{ij})$ is the probability of tc_{ij}. Next, we compute the ratio of the each candidate's score over the highest candidate's score as follows: $ScoreRatio(tc_{ij}) =$ MI score$(tc_{ij}|Q)$/MI score$(tc_{ih}|Q)$, where tc_{ih} is the candidate with highest MI score from the qt_i. If the candidate's score ratio is below the threshold τ_{MI}, the candidate will be discarded.

Here, we use the above example to illustrate the term disambiguation mechanism. For the given English term "Kosovo", the MI scores of "科索夫", "科索伏", and "科索沃" are computed; "科索伏" achieves the highest score, while the score ratio of the other two candidates are much lower than the threshold. Thus, only "科索伏" is treated as Kosovo's translation and used to build the final Chinese query to perform the IR.

4.3 Indexing and Retrieval Model

We use the Lucene information retrieval engine to index all documents and the bigram index based on Chinese characters. The Okapi BM25 function (Robertson et al., 1996) is used to score

a retrieved document's relevance. In addition, we employ the following document re-ranking function (Yang et al., 2007):

$$\sqrt{\frac{(\sum_{i=1}^{K} df(t,d_i) \times f(i))/K}{DF(t,C)/R}} \times \sqrt{|t|}$$

$$df(t,d_i) = \begin{cases} 1 & t \in d_i \\ 0 & t \notin d_i \end{cases},$$

where d_i is the ith document; R is the total number of documents in the collection C; $DF(t,C)$ is the number of documents containing a term t in C; and $|t|$ is t's length, $f(i) = \frac{1}{sqrt(i)}$.

5 Evaluation and Analysis

To evaluate our KCIR system, we use the topic and document collections of the NTCIR-5 CLIR tasks (Kishida et al., 2005). The document collection is the Chinese Information Retrieval Benchmark (CIRB) 4.0, which contains news articles published in four Taiwanese newspapers from 2000 to 2001. The topics have four fields: title, description, narration, and concentrate words. We use 50 topics provided by NTCIR-5 and use the title field as the input query because it is similar to queries input to search engines.

We construct five runs as follows:

- **Baseline**: using a Korean-Chinese dictionary-based translation.
- **Baseline+Extended Dictionaries only**: the baseline system plus the extended dictionaries translation.
- **Baseline+NET Methods**: the baseline system plus our NET methods, namely, Wikipedia, Naver people search, and the pattern-based method.
- **Google Translation**: using the Google translation tool.
- **Chinese monolingual**: using the Chinese versions of the topics given by NTCIR.

We use the Mean Average Precision (MAP) and Recall (Saracevic et al., 1988) to evaluate the performance of IR. NTCIR provides two kinds of relevance judgments: Rigid and Relax. A document is rigid-relevant if it is highly relevant to the topic; and relax-relevant if it is highly relevant or partially relevant to the topic.

Table 1: Evaluation Results

Run	MAP		Recall	
	Rigid	Relax	Rigid	Relax
Baseline	0.0553	0.0611	0.2202	0.2141
Baseline+extended dictionaries	0.1573	0.1751	0.5706	0.5489
Baseline+NET	**0.2576**	**0.2946**	**0.7255**	**0.7103**
Google translation	0.1340	0.1521	0.5254	0.5149
Chinese mono	0.2622	0.3019	0.7705	0.7452

Table 1 shows that our method improves KCIR substantially. Our method's performance is about five times better than that of the baseline system and very close to that of Chinese monolingual IR. Wikipedia translation improves the performance, but not markedly because Wikipedia cannot cover some NEs. Google translation is not very satisfactory either, since many NEs cannot be translated correctly.

To evaluate our NE translation method, we create two additional datasets. The first dataset contains all the 30 topics with NEs in NTCIR-5. To further investigate the effectiveness of our method for queries containing person names, which are the most frequent NEs, we construct a second dataset containing 16 topics with person names in NTCIR-5. We compare the performance of our method on KCIR with that of Chinese monolingual IR on these two datasets. The results are shown in Tables 2 and 3.

5.1 Effectiveness of Extended Dict

We adopt two online dictionaries to extend our bilingual dictionaries: Wikipedia and Naver people search engine. Wikipedia is an effective tool for translating well-known NEs. In the test topics, NEs like "김정일"(Kim Jong-il, North Korea's leader), "탈리반"(Taliban), "해리포터"(Harry Potter) and "한나라당"(Great National Party in South Korea) are all translated correctly by Wikipedia.

We observe that the most difficult cases in Korean-Chinese person name translation, especially Japanese and non-CJK person names, can

be successfully translated by the Naver people search engine. For example, "코엔"(William Cohen, the ex-Secretary of Defense of the U.S.) and "이치로"(Ichiro Suzuki, a Japanese baseball player). The major advantage of the Naver people search engine is it can can provide the original names written in Chinese characters.

According to our evaluation, the extended dictionaries improve the IR performance of the baseline system about threefold. It shows that the extended dictionaries can translate part of Korean NEs into Chinese. However, there are still many NEs that the extended dictionaries cannot cover.

5.2 Effectiveness of Patterns

In our method, we employ automatically learned patterns to extract translations for the remaining NEs not covered by the offline or online dictionaries. For example, we can extract Chinese translations for "오키나와"(Okinawa, in Japan) by using K-C translation patterns. Most non-CJK NEs can be translated correctly by using the K-E translation patterns. For example, "제니퍼 카프리아티"(Jennifer Capriati), "탄저"(anthrax), and "광우병"(mad cow disease) can be extracted from Google snippets effectively by our translation patterns.

Although our method translates some NEs into English first and then into Chinese in an indirect manner, it is very effective because the non-CJK NEs in Korean are mainly from English. In fact, 16 of the 17 NEs can be successfully translated by the two stage translation method that employs two types of translation patterns: K-E and E-C.

5.3 Effectiveness Analysis of NET

As shown in Table 2, for topics with NEs, the rigid MAP of our method is very close to that of Chinese monolingual IR, while the relax MAP of our method is even better than that of Chinese monolingual IR. We observe that 26 of the 31 NEs in the topics are successfully translated into Chinese. These results demonstrate that our hybrid method comprising the extended dictionaries and translation patterns can deal with Korean-Chinese NE translation effectively and improve the performance of IR substantially.

Table 2: Results on Topics with NEs

Run	MAP		Recall	
	Rigid	Relax	Rigid	Relax
NET	0.2700	0.3385	0.7565	0.7578
Chinese	0.2746	0.3273	0.7922	0.7846

Note that, our method can extract more possible Chinese translations, which is similar to query expansion. For non-CJK NEs, there may exist several Chinese transliterations that are actually used in Chinese, especially for the person names. Take "Tito"for example; its six common Chinese transliterations, namely, "迪托"(di-tuo), "蒂托"(di-tuo), "帝托"(di-tuo), "提托"(ti-tuo), and "狄托"(di-tuo) can be extracted. With our method, the rigid MAP of this topic achieves 0.8361, which is much better than that of the same topic in the Chinese monolingual run (0.4459) because the Chinese topic has only one transliteration "帝托"(di-tuo). This is the reason that our method outperforms the Chinese monolingual run in topics with NEs.

5.4 Error Analysis

NEs that cannot be translated correctly can be divided into two categories. The first contains names not selected as NE candidates. The Japanese person name "후지모리" (Alberto Fujimori, Peru's ex-president) is in this category. For the name "후지모리" (Fujimori), the first two characters "후지" (hind legs) and the last two characters "모리" (profiting) are all Sino-Korean words, so it is regarded as a compound word, not an NE. The other category contains names with few relevant web pages, like the non-CJK names "안토니오 토디" (Antonio Toddy).

The other problem is that our method can translate the Korean NEs into correct Chinese translations, but not the translation used in the CIRB 4.0 news collection. For example, "쿠르스크" (Kursk) is translated into "庫爾斯克" (Ku-er-si-ke) correctly, but only the transliteration "科斯克" (Ke-si-ke) is used in CIRB 4.0. In this situation, the extracted translation cannot improve the performance of the KCIR.

Table 3: Results on Topics with Person Names

Run	MAP		Recall	
	Rigid	Relax	Rigid	Relax
NET	0.2730	0.3274	0.7146	0.7299
Chinese	0.2575	0.3169	0.7513	0.7708

6 Conclusion

In this paper, we have considered the difficulties that arise in translating NEs from Korean to Chinese for IR. We propose a hybrid method for K-C NET that exploits an extended dictionarie containing Wikipedia and the Naver people search engine, combined with the translation patterns automatically learned from the search results of the Google search engine. To evaluate our method, we use the topics and document collection of the NTCIR-5 CLIR task. Our method's performance on KCIR is over five times better than that of the baseline configuration with only an offline dictionary-based translation module. Moreover, its overall MAP score is up to 0.2986, and its MAP on the NE topics is up to 0.3385 which is even better than that of the Chinese monolingual IR system. The proposed method can translate NEs that originated in the Chinese, Japanese, Korean, and non-CJK languages and improve the performance of KCIR substantially. Our NET method is not language-specific; therefore, it can be applied to the other CLIR systems beside K-C IR.

References

Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 400–408.

Hsin-Hsi Chen, Sheng-Jie Huang, Yung-Wei Ding, and Shih-Cbung Tsai. 1998. Proper name translation in cross-language information retrieval. *Proceedings of 17th COLING and 36th ACL*, pages 232–236.

Kazuaki Kishida, Kuang hua Chen, Sukhoon Lee, Kazuko Kuriyama, Noriko Kando, Hsin-Hsi Chen, and Sung Hyon Myaeng. 2005. Overview of clir task at the fifth ntcir workshop. *Proceedings of the Fifth NTCIR Workshop*.

Juhee Lee. 2003. Loadword phonology revisted: Implications of richness of the base for the analysis of loanwords input. *Explorations in Korean Language and Linguistics*, pages 361–375.

Mira Oh. 2003. English fricatives in loanword adaption. *Explorations in Korean Language and Linguistics*, pages 471–487.

S.E. Robertson, S. Walker, MM Beaulieu, M. Gatford, and A. Payne. 1996. Okapi at trec-4. *Proceedings of the Fourth Text Retrieval Conference*, pages 73–97.

Tefko Saracevic, Paul Kantor, Alice Y. Chamis, and Donna Trivison. 1988. A study of information seeking and retrieving. *Journal of the American Society for Information Science*, 39(3):161–176.

L. Yang, D. Ji, and M. Leong. 2007. Document reranking by term distribution and maximal marginal relevance for chinese information retrieval. *Information Processing and Management: an International Journal*, 43(2):315–326.

Bootstrapping Both Product Features and Opinion Words from Chinese Customer Reviews with Cross-Inducing[1]

Bo Wang
Institute of Computational Linguistics
Peking University
Beijing, 100871, China
wangbo@pku.edu.cn

Houfeng Wang
Institute of Computational Linguistics
Peking University
Beijing, 100871, China
wanghf@pku.edu.cn

Abstract

We consider the problem of identifying product features and opinion words in a unified process from Chinese customer reviews when only a much small seed set of opinion words is available. In particular, we consider a problem setting motivated by the task of identifying product features with opinion words and learning opinion words through features alternately and iteratively. In customer reviews, opinion words usually have a close relationship with product features, and the association between them is measured by a revised formula of mutual information in this paper. A bootstrapping iterative learning strategy is proposed to alternately both of them. A linguistic rule is adopted to identify low-frequent features and opinion words. Furthermore, a mapping function from opinion words to features is proposed to identify implicit features in sentence. Empirical results on three kinds of product reviews indicate the effectiveness of our method.

1 Introduction

With the rapid expansion of network application, more and more customer reviews are available online, which are beneficial for product merchants to track the viewpoint of old customers and to assist potential customers to purchase products. However, it's time-consuming to read all reviews in person. As a result, it's significant to mine customer reviews automatically and to provide users with opinion summary.

In reality, product features and opinion words play the most important role in mining opinions of customers. One customer review on some cell phone is given as follows:

(a) "**外型**漂亮，**屏幕**大，**拍照效果**好。"(The **appearance** is beautiful, the **screen** is big and the **photo effect** is OK.)

Product features are usually nouns such as "外型" (appearance) and "屏幕" (screen) or noun phrases such as "拍照效果" (photo effect) expressing which attributes the customers are mostly concerned. Opinion words (opword is short for "opinion word") are generally adjectives used to express opinions of customers such as "漂亮" (beautiful), "大" (big) and "好" (well). As the core part of an opinion mining system, this paper is concentrated on identifying both product features and opinion words in Chinese customer reviews.

There is much work on feature extraction and opinion word identification. Hu and Liu (2004) makes use of association rule mining (Agrawal and Srikant, 1994) to extract frequent features, the surrounding adjectives of any extracted feature are considered as opinion words. Popescu and Etzioni (2005) has utilized statistic-based point-wise mutual information (PMI) to extract product features. Based on the association of opinion words with product features, they take the advantage of the syntactic dependencies computed by the MINIPAR parser (Lin, 1998) to identify opinion words. Tur-

[1] Supported by National Natural Science Foundation of China under grant No.60675035 and Beijing Natural Science Foundation under grant No.4072012

ney (2002) applied a specific unsupervised learning technique based on the mutual in-formation between document phrases and two seed words "excellent" and "poor".

Nevertheless, in previous work, identifying product features and opinion words are always considered two separate tasks. Actually, most product features are modified by the surrounding opinion words in customer reviews, thus they are highly context dependent on each other, which is referred to as context-dependency property henceforth. With the co-occurrence characteristic, identifying product features and opinion words could be combined into a unified process. In particular, it is helpful to identify product features by using identified opinion words and vice versa. That implies that such two subtasks can be carried out alternately in a unified process. Since identifying product features are induced by opinion words and vice versa, this is called cross-inducing.

As the most important part of a feature-based opinion summary system, this paper focuses on learning product features and opinion words from Chinese customer reviews. Two sub-tasks are involved as follows:

Identifying features and opinion words: Resorting to context-dependency property, a bootstrapping iterative learning strategy is proposed to identify both of them alternately.

Identifying implicit features: Implicit features occur frequently in customer reviews. An implicit feature is defined as a feature that does not appear in an opinion sentence. The association between features and opinion words calculated with the revised mutual information is used to identify implicit features.

This paper is sketched as follows: Section 2 describes the approach in detail; Experiment in section 3 indicates the effectiveness of our approach. Section 4 presents related work and section 5 concludes and presents the future work.

2 The Approach

Figure 1 illustrates the framework of an opinion summary framework, the principal parts related to this paper are shown in bold. The first phase "identifying features and opinion words", works iteratively to identify features with the opinion words identified and learn opinion words through the product features identified alternately. Then, one linguistic rule is used to identify low-frequent features and opinion words. After that, a mapping function is designed to identify implicit features.

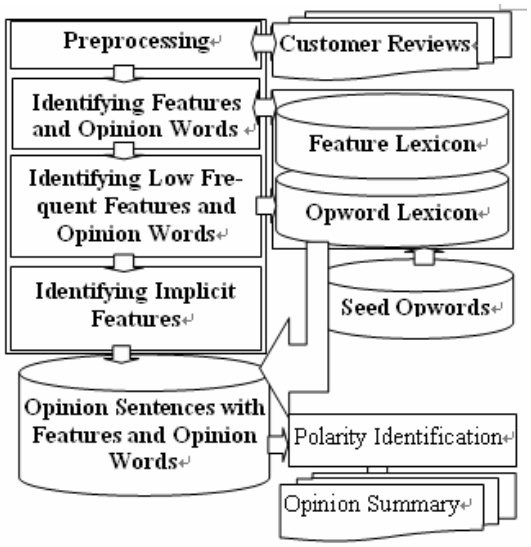


Figure 1. The framework of an opinion summary system

2.1 Iterative Learning Strategy

Product features and opinion words are highly context-dependent on each other in customer reviews, i.e., the feature "机身" (body) for digital camera often co-occur with some opinion words such as "大" (big) or "小巧" (delicate) while the feature "性价比" (the proportion of performance to price) often co-occurs with the opinion word "高" (high).

Product features can be identified resorting to the surrounding opinion words identified before and vice versa. A bootstrapping method that works iteratively is proposed in algorithm 1.

Algorithm 1 works as follows: given the seed opinion words and all the reviews, all noun phrases (noun phrases in the form "noun+") form *CandFeaLex* (the set of feature candidates) and all adjectives compose of *CandOpLex* (the set of the candidates of opinion words). The sets *ResFeaLex* and *ResOpLex* are used to store final features and opinion words. Initially, *ResFeaLex* is set empty while *ResOpLex* is composed of all the seed opinion words. At each iterative step, each feature *candidate* in *CandFeaLex* is scored by its context-dependent association with each *opword* in *ResOpLex*, the candidate whose score is above the prespecified threshold $Threshold_{feature}$ is added to *Res*

Algorithm 1. Bootstrap learning product features and opinion words with cross-inducing

Bootstrap-Learning (ReviewData, SeedOpLex, $\text{Threshold}_{\text{feature}}$, $\text{Threshold}_{\text{opword}}$)

```
1  Parse(ReviewData);
2  ResFeaLex = {}, ResOpLex = SeedOpLex;
3  CandFeaLex = all noun phrases in ReviewData;
4  CandOpLex = all adjectives in ReviewData;
5  while (CandFeaLex≠{} && CanOpLex≠{})
6     do for each candfea∈CandFeaLex
7                do for each opword∈ResOpLex
8             do calculate RMI(candfea,opword) with ReviewData;
9          score(canfea)=Σ_{opword∈ResOpLex}RMI(candfea,opword)/|ResOpLex|;
10     sort CandFeaLex by score;
11     for each candfea∈CandFeaLex
12        do if (score(candfea)> Threshold_feature)
13                          then  ResFeaLex=ResFeaLex+{candfea};
14                   CanFeaLex=CandFeaLex – {candfea};
15     for each candop∈CandOpLex
16        do for each feature∈ResFeaLex
17                      do calculate RMI(candop,feature) with D;
18        score(candop)=Σ_{feature∈ResFeaLex}RMI(feature,candop)/|ResFeaLex| ;
19     sort CandOpLex by score;
20     for each candop∈CandOpLex
21                do if (score (candop)>Threshold_opword)
22        then ResOpLex=ResOpLex+{candop };
23           CanOpLex=CandOpLex – {candop};
24     if (neither candfea and candop is learned) then break;
25 return ResFeaLex, ResOpLex;
```

FeaLex and subtracted from *CandFeaLex*. Similarly, opinion words are processed in this way, but the scores are related to features in *ResFeaLex*. The iterative process continues until neither *ResFeaLex* nor *ResOpLex* is altered. Any feature candidate and opinion word candidate, whose relative distance in sentence is less than or equal to the specified window size *Minimum-Offset*, are regarded to co-occur with each other. The association between them is calculated by the revised mutual information denoted by *RMI*, which will be described in detail in the following section and employed to identify implicit features in sentences.

2.2 Revised Mutual Information

In customer reviews, features and opinion words usually co-occur frequently, features are usually modified by the surrounding opinion words. If the absolute value of the relative distance in a sentence for a feature and an opinion word is less than *Minimum-Offset*, they are considered context-dependent.

Many methods have been proposed to measure the co-occurrence relation between two words such as χ^2 (Church and Mercer,1993) , mutual information (Church and Hanks, 1989; Pantel and Lin, 2002), t-test (Church and Hanks, 1989), and log-likelihood (Dunning,1993). In this paper a revised formula of mutual information is used to measure the association since mutual information of a low-frequency word pair tends to be very high.

Table 1 gives the contingency table for two words or phrases w_1 and w_2, where A is the number of reviews where w_1 and w_2 co-occur; B indicates the number of reviews where w_1 occurs but does not co-occur with w_2; C denotes the number of reviews where w_2 occurs but does not co-occur with w_1; D is number of reviews where neither w_1 nor w_2 occurs; $N = A + B + C + D$.

With the table, the revised formula of mutual information is designed to calculate the association of w_1 with w_2 as formula (1).

	$\mathbf{w_2}$	$\mathbf{\sim w_2}$
$\mathbf{w_1}$	A	B
$\mathbf{\sim w_1}$	C	D

Table 1: Contingency table

$$RMI(w_1, w_2) = freq(w_1, w_2) \times \log \frac{p(w_1, w_2)}{p(w_1) \bullet p(w_2)}$$

$$= A \times \log \frac{N \times A}{(A+B) \times (A+C)} \quad (1)$$

2.3 Identifying Low-Frequent Features and Opinion Words

In Chinese reviews, one linguistic rule "noun+ adverb* adjective+" occurs frequently and most of the instances of the rule are used to express positive or negative opinions on some features, i.e., "机身/noun 比较/adverb 小巧/adjective" (The body is rather delicate) , where each Chinese word and its part-of-speech is separated by the symbol "/".

Intuitively, this linguistic rule can be used to improve the output of the iterative learning. For each instance of the rule, if "noun+" exists in *ResFeaLex*, the "adjective" part would be added to *ResOpLex*, and if "adjective+" exists in *ResOpLex*, the noun phrase "noun+" part will be added to *ResFeaLex*. After that, most low-frequent features and opinion words will be recognized.

2.4 Identifying Implicit Features

The context-dependency property indicates the context association between product features and opinion words. As a result, with the revised mutual information, the implicit features can be deduced from opinion words. A mapping function *f: opword → feature* is used to deduce the mapping feature for opword , where *f(opword)* is defined as the feature with the largest association with opinion word.

If an opinion sentence contains opinion words, but it does not have any explicit features, the mapping function *f: opword → feature* is employed to generate the implicit feature for each opinion word and the feature is considered as an implicit feature in the opinion sentence. Two instances are given in (b) and (c), where the implicit features are inserted in suitable positions and they are separated in parentheses. Since *f* ("漂亮" (beautiful)) = "外观" (appearance) and *f* ("时尚" (fashionable)) = "外观" (appearance), "外观" (appearance) is an implicit feature in (b). Similarly, the implicit features in (c) are "性能" (performance) and "图像" (picture).

(b) (**外观**)漂亮而且(**外观**)时尚。It's (**appearance**) beautiful and (**appearance**) fashionable.

(c) (**性能**)很稳定，而且(**图像**)很清晰。It's (**performance**) very stable and (**picture**) very clear.

3 Experiment

3.1 Data Collection

We have gathered customer reviews of three kinds of electronic products from http://it168.com: digital camera, cell-phone and tablet. The first 300 reviews for each kind of them are downloaded. One annotator was asked to label each sentence with product features (including implicit features) and opinion words. The annotation set for features and opinion words are shown in table 2.

Product Name	No. of Features	No. of Opinion Words
digital camera	135	97
cell-phone	155	125
tablet	96	83

Table 2. Annotation set for product features and opinion words

Unlike English, Chinese are not separated by any symbol. Therefore, the reviews are tokenized and tagged with part-of-speech by a tool ICTCLAS[2].One example of the output of this tool is as (d).

(d) 开机/n 速度/n 还/d 满/d 快/a ，/w 镜头/n 保护盖/n 拉开/v 就/d 可以/v 进入/v 拍摄/n 状态/n ，/w 模式/n 选择/vn 切换/vn 也/d 很/d 方便/a 。/w

The seed opinion words employed in the iterative learning are: "清晰" (clear), "快" (quick), "白" (white), "差劲" (weak). "好" (good), "不错" (good), "高" (high), "小" (little), "多" (many), "长" (long). Empirically, $Threshold_{feature}$ and $Threshold_{opword}$ in Algorithm 1 is set to 0.2, *Minimum-Offset* is set to 4.

[2] http://www.nlp.org.cn

Product Name	On Set			On Sentence		
	Precision	Recall	F-Score	Precision	Recall	F-Score
digital camera	64.03%	45.92%	53.49%	46.62%	65.72%	54.55%
cell-phone	54.43%	43.87%	48.58%	34.17%	55.15%	42.19%
tablet	51.45%	59.38%	55.13%	41.39%	60.21%	49.06%
average	**56.64%**	**49.72%**	**52.40%**	**40.73%**	**60.36%**	**48.60%**

Table 3. Evaluation of apriori algorithm

Type	Product Name	On Set			On Sentence		
		Precision	Recall	F-Score	Precision	Recall	F-score
feature	**digital camera**	73.57%	54.81%	62.82%	55.80%	68.69%	61.58%
		78.20%	73.33%	75.69%	54.71%	70.80%	63.49%
	cell-phone	80.92%	45.81%	58.50%	47.31%	58.59%	52.35%
		82.30%	66.46%	73.53%	49.22%	61.63%	54.73%
	tablet	72.73%	57.29%	64.09%	49.79%	61.03%	54.84%
		77.99%	73.96%	75.92%	52.54%	64.43%	57.88%
	average	**75.74%**	**52.64%**	**61.80%**	**50.97%**	**62.77%**	**56.26%**
		79.50%	**71.25%**	**75.05%**	**52.16%**	**65.62%**	**58.70%**
opword	**digital camera**	89.02%	38.02%	53.28%	72.35%	50.24%	59.30%
		87.31%	60.94%	71.78%	69.40%	85.28%	76.53%
	cell-phone	87.95%	30.80%	45.63%	66.44%	42.84%	52.09%
		88.49%	51.90%	65.43%	63.14%	79.51%	70.39%
	tablet	77.94%	30.64%	43.98%	61.30%	42.69%	50.34%
		80.73%	50.87%	62.41%	63.92%	81.02%	71.46%
	average	**84.97%**	**33.15%**	**47.63%**	**66.70%**	**45.26%**	**53.91%**
		85.51%	**54.57%**	**66.54%**	**65.49%**	**81.94%**	**72.79%**

Table 4. Evaluation of iterative learning (the upper) and the combination of iterative learning and the linguistic rule (the lower).

3.2 Evaluation Measurement

As Hu and Liu (2004), the features mined form the result set while the features in the manually annotated corpus construct the answer set. With the two sets, precision, recall and f-score are used to evaluate the experiment result on set level.

In our work, the evaluation is also conducted on sentence for three factors: Firstly, each feature or opinion word may occur many times in reviews but it just occurs once in the corresponding answer set; Secondly, implicit features should be evaluated on sentence; Besides, to generate an opinion summary, the features and the opinion words should be identified for each opinion sentence.

On sentence, the features and opinion words identified for each opinion sentence are compared with the annotation result in the corresponding sentence. Precision, recall and f-score are also used to measure the performance.

3.3 Evaluation

Hu and Liu (2004) have adopted associate rule mining to mine opinion features from customer reviews in English. Since the original corpus and source code is not available for us, in order to make comparison with theirs, we have re-implemented their algorithm, which is denoted as apriori method as follows. To be pointed out is that, the two pruning techniques proposed in Hu and Liu (2004): compactness pruning and redundancy pruning, were included in our experiment. The evaluation on our test data is listed in table 3. The row indexed by average denotes the average performance of the corresponding column and each entry in it is bold.

Table 4 shows our testing result on the same data, the upper value in each entry presents the result for iterative learning strategy while the lower values denote that for the combination of iterative learning and the linguistic rule. The average row

shows the average performance for the corresponding columns and each entry in the row is shown in bold.

On feature, the average precision, recall and f-score on set or sentence increase according to the order apriori < iterative < ite+rule, where apriori indicates Hu and Liu's method, iterative represents iterative strategy and iterative+rule denotes the combination of iterative strategy and the linguistic rule. The increase range from apriori to iterative+rule of f-score on set gets to 22.65% while on sentence it exceeds 10%. The main reason for the poor performance on set for apriori is that many common words such as "电脑" (computer), "中国" (China) and "时间" (time of use) with high frequency are extracted as features. Moreover, the poor performance on sentence for apriori method is due to that it can't identify implicit features. Furthermore, the increase in f-score from iterative to ite+rule on set and on sentence shows the performance can be enhanced by the linguistic rule.

Table 4 also shows that the performance in learning opinion words has been improved after the linguistic rule has been used. On set, the average precision increases from 84.97% to 85.51% while the average recall from 33.15% to 54.57%. Accordingly, the average f-score increase significantly by about 18.91%.

On sentence, although there is a slow decrease in the average precision, there is a dramatic increase in the average recall, thus the average f-score has increased from 53.91% to 72.79%. Furthermore, the best f-score (66.54%) on set and the best f-score (72.79%) on sentence indicate the effectiveness of *ite+rule* on identifying opinion words.

4 Related Work

Our work is much related to Hu's system (Hu and Liu,2004), in which association rule mining is used to extract frequent review noun phrase as features. After that, two pruning techniques: compactness pruning and redundancy pruning, are utilized. Frequent features are used to find potential opinion words (adjectives) and WordNet synonyms/antonyms in conjunction with a set of seed words are used in order to find actual opinion words. Finally, opinion words are used to extract associated infrequent features. The system only extracts explicit features. Our work differs from hers at two aspects: (1) their method can't identify implicit features which occur frequently in opinion sentences; (2) Product features and opinion words are identified on two separate steps in Hu's system but they are learned in a unified process here and induced by each other in this paper.

Popescu and Etzioni (2005) has used web-based point-wise mutual information (PMI) to extract product features and use the identified features to identify potential opinion phrases with co-occurrence association. They take advantage of the syntactic dependencies computed by the MINIPAR parser. If an explicit feature is found in a sentence, 10 extraction rules are applied to find the heads of potential opinion phrases. Each head word together with its modifier is returned as a potential opinion phrase. Our work is different from theirs on two aspects: (1) Product features and opinion words are identified separately but they are learned simultaneously and are boosted by each other here. (2) They have utilized a syntactic parser MINIPAR, but there's no syntactic parser available in Chinese, thus the requirement of our algorithm is only a small seed opinion word lexicon. Although co-occurrence association is used to derive opinion words from explicit features in their work, the way how co-occurrence association is represented is different. Besides, the two sub-tasks are boosted by each other in this paper.

On identifying opinion words, Morinaga et al (2002)has utilized information gain to extract classification features with a supervised method; Hatzivassiloglou and Wiebe (1997) used textual junctions such as "fair and legitimate" or "simplistic but well-received" to separate similarity- and oppositely-connoted words; Other methods are present in (Riloff et al, 2003; Riloff and Wiebe, 2003; Gamon and Aue, 2005; Wilson et al, 2006) The principal difference from previous work is that, they have considered extracting opinion words as a separate work but we have combined identifying features and opinion words in a unified process. Besides, the opinion words are identified for sentences but in their work they are identified for reviews.

5 Conclusion

In this paper, identifying product features and opinion words are induced by each other and are combined in a unified process. An iterative learn-

ing strategy based on context-dependence property is proposed to learn product features and opinion words alternately, where the final feature lexicon and opinion word lexicon are identified with very few knowledge (only ten seed opinion words) and augmented by each other alternately. A revised formula of mutual information is used to calculate the association between each feature and opinion word. A linguistic rule is utilized to recall low-frequent features and opinion words. Besides, a mapping function is designed to identify implicit features in sentence. In addition to evaluating the result on set, the experiment is evaluated on sentence. Empirical result indicates that the performance of iterative learning strategy is better than apriori method and that features and opinion words can be identified with cross-inducing effectively. Furthermore, the evaluation on sentence shows the effectiveness in identifying implicit features.

In future, we will learn the semantic orientation of each opinion word, calculate the polarity of each subjective sentence, and then construct a feature-based summary system.

References

Ana Maria Popescu and Oren Etzioni. 2005. *Extracting Product Features and Opinions from Reviews*. Proceedings of HLT-EMNLP (2005)

De-Kang Lin. 1998. *Dependency-Based Evaluation of MINIPAR*. In:Proceedings of the Workshop on the Evaluation of Parsing Systems, Granada, Spain, 1998, 298～312

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. *Learning Subjective Nouns Using Extraction Pattern Bootstrapping*. Seventh Conference on Natural Language Learning (CoNLL-03). ACL SIGNLL. Pages 25-32.

Ellen Riloff and Janyce Wiebe. 2003. *Learning Extraction Patterns for Subjective Expressions*. Conference on Empirical Methods in Natural Language Processing (EMNLP-03). ACL SIGDAT. 2003, 105-112.

Kenneth Ward Church and Robert L. Mercer. 1993. *Introduction to the special issue on computational linguistics using large corpora.* Computational Linguistics 19:1-24

Kenneth Ward Church and Patrick Hanks. 1989. *Word Association Norms, Mutual Information and Lexicography.* Proceedings of the 26th Annual Conference of the Association for Computational Linguistics(1989).

Michael Gamon and Anthony Aue. 2005. *Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms.* In :ACL 2005 Workshop on Feature Engineering,2005.

Minqing Hu and Bing Liu. 2004. *Mining Opinion Features in Customer Reviews.* Proceedings of Nineteeth National Conference on Artificial Intellgience (AAAI-2004), San Jose, USA, July 2004.

Patrick Pantel and Dekang Lin. 2002. *Document Clustering with Committees.* In Proceedings of ACM Conference on Research and Development in Information Retrieval (SIGIR-02). pp. 199-206. Tampere, Finland.

Peter D. Turney. 2002. *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. ACL 2002: 417-424

Rakesh Agrawal and Ramakrishan Srikant. 1994. *Fast algorithm for mining association rules.* VLDB'94, 1994.

Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. *Mining Product Reputations on the WEB*, Proceedings of 8th ACM SIGKDD International Conference on Knowledge. Discover and Data Mining, (2002) 341-349

Ted Dunning. 1993. *Accurate methods for the statistics of surprise and coincidence.* Computational Linguistics 19:61-74

Theresa Wilson , Janyce Wiebe, and Rebecca Hwa. 2006. *Recognizing strong and weak opinion clauses*. Computational Intelligence 22 (2): 73-99.

Learning to Shift the Polarity of Words for Sentiment Classification

Daisuke Ikeda† **Hiroya Takamura**‡ **Lev-Arie Ratinov**†† **Manabu Okumura**‡

†Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology
ikeda@lr.pi.titech.ac.jp

††Department of Computer Science, University of Illinois at Urbana-Champaign
ratinov2@uiuc.edu

‡Precision and Intelligence Laboratory, Tokyo Institute of Technology
{takamura,oku}@pi.titech.ac.jp

Abstract

We propose a machine learning based method of sentiment classification of sentences using word-level polarity. The polarities of words in a sentence are not always the same as that of the sentence, because there can be polarity-shifters such as negation expressions. The proposed method models the polarity-shifters. Our model can be trained in two different ways: word-wise and sentence-wise learning. In sentence-wise learning, the model can be trained so that the prediction of sentence polarities should be accurate. The model can also be combined with features used in previous work such as bag-of-words and n-grams. We empirically show that our method almost always improves the performance of sentiment classification of sentences especially when we have only small amount of training data.

1 Introduction

Due to the recent popularity of the internet, individuals have been able to provide various information to the public easily and actively (e.g., by weblogs or online bulletin boards). The information often includes opinions or sentiments on a variety of things such as new products. A huge amount of work has been devoted to analysis of the information, which is called *sentiment analysis*. The sentiment analysis has been done at different levels including words, sentences, and documents. Among them, we focus on the sentiment classification of sentences, the task to classify sentences into "positive" or "negative", because this task is fundamental and has a wide applicability in sentiment analysis. For example, we can retrieve individuals' opinions that are related to a product and can find whether they have the positive attitude to the product.

There has been much work on the identification of sentiment polarity of words. For instance, "beautiful" is positively oriented, while "dirty" is negatively oriented. We use the term *sentiment words* to refer to those words that are listed in a predefined polarity dictionary. Sentiment words are a basic resource for sentiment analysis and thus believed to have a great potential for applications. However, it is still an open problem how we can effectively use sentiment words to improve performance of sentiment classification of sentences or documents.

The simplest way for that purpose would be the majority voting by the number of positive words and the number of negative words in the given sentence. However, the polarities of words in a sentence are not always the same as that of the sentence, because there can be polarity-shifters such as negation expressions. This inconsistency of word-level polarity and sentence-level polarity often causes errors in classification by the simple majority voting method. A manual list of polarity-shifters, which are the words that can shift the sentiment polarity of another word (e.g., negations), has been suggested. However, it has limitations due to the diversity of expressions.

Therefore, we propose a machine learning based method that models the polarity-shifters. The model can be trained in two different ways: *word-wise*

and *sentence-wise*. While the word-wise learning focuses on the prediction of polarity shifts, the sentence-wise learning focuses more on the prediction of sentence polarities. The model can also be combined with features used in previous work such as bag-of-words, n-grams and dependency trees. We empirically show that our method almost always improves the performance of sentiment classification of sentences especially when we have only small amount of training data.

The rest of the paper is organized as follows. In Section 2, we briefly present the related work. In Section 3, we discuss well-known methods that use word-level polarities and describe our motivation. In Section 4, we describe our proposed model, how to train the model, and how to classify sentences using the model. We present our experiments and results in Section 5. Finally in Section 6, we conclude our work and mention possible future work.

2 Related Work

Supervised machine learning methods including Support Vector Machines (SVM) are often used in sentiment analysis and shown to be very promising (Pang et al., 2002; Matsumoto et al., 2005; Kudo and Matsumoto, 2004; Mullen and Collier, 2004; Gamon, 2004). One of the advantages of these methods is that a wide variety of features such as dependency trees and sequences of words can easily be incorporated (Matsumoto et al., 2005; Kudo and Matsumoto, 2004; Pang et al., 2002). Our attempt in this paper is not to use the information included in those substructures of sentences, but to use the word-level polarities, which is a resource usually at hand. Thus our work is an instantiation of the idea to use a resource on one linguistic layer (e.g., word level) to the analysis of another layer (sentence level).

There have been some pieces of work which focus on multiple levels in text. Mao and Lebanon (2006) proposed a method that captures local sentiment flow in documents using isotonic conditional random fields. Pang and Lee (2004) proposed to eliminate objective sentences before the sentiment classification of documents. McDonald et al. (2007) proposed a model for classifying sentences and documents simultaneously. They experimented with joint classification of subjectivity for sentence-level, and sentiment for document-level, and reported that their model obtained higher accuracy than the standard document classification model.

Although these pieces of work aim to predict not sentence-level but document-level sentiments, their concepts are similar to ours. However, all the above methods require annotated corpora for all levels, such as both subjectivity for sentences and sentiments for documents, which are fairly expensive to obtain. Although we also focus on two different layers, our method does not require such expensive labeled data. What we require is just sentence-level labeled training data and a polarity dictionary of sentiment words.

3 Simple Voting by Sentiment Words

One of the simplest ways to classify sentences using word-level polarities would be a majority voting, where the occurrences of positive words and those of negative words in the given sentence are counted and compared with each other. However, this majority voting method has several weaknesses. First, the majority voting cannot take into account at all the phenomenon that the word-level polarity is not always the same as the polarity of the sentence. Consider the following example:

> I have not had any distortion problems with this phone and am more pleased with this phone than any I've used before.

where negative words are underlined and positive words are double-underlined. The example sentence has the positive polarity, though it locally contains negative words. The majority voting would misclassify it because of the two negative words.

This kind of inconsistency between sentence-level polarity and word-level polarity often occurs and causes errors in the majority voting. The reason is that the majority voting cannot take into account negation expressions or adversative conjunctions, e.g., "I have not had any ..." in the example above. Therefore, taking such polarity-shifting into account is important for classification of sentences using a polarity dictionary. To circumvent this problem, Kennedy and Inkpen (2006) and Hu and Liu (2004) proposed to use a manually-constructed list of polarity-shifters. However, it has limitations due to the diversity of expressions.

Another weakness of the majority voting is that it cannot be easily combined with existing methods that use the n-gram model or tree structures of the sentence as features. The method we propose here can easily be combined with existing methods and show better performance.

4 Word-Level Polarity-Shifting Model

We assume that when the polarity of a word is different from the polarity of the sentence, the polarity of the word is shifted by its context to adapt to the polarity of the sentence. Capturing such polarity-shifts will improve the classification performance of the majority voting classifier as well as of more sophisticated classifiers.

In this paper, we propose a word polarity-shifting model to capture such phenomena. This model is a kind of binary classification model which determines whether the polarity is shifted by its context. The model assigns a score $s_{shift}(x, S)$ to the sentiment word x in the sentence S. If the polarity of x is shifted in S, $s_{shift}(x, S) > 0$. If the polarity of x is not shifted in S, $s_{shift}(x, S) \leq 0$. Let $\boldsymbol{w}$ be a parameter vector of the model and $\boldsymbol{\phi}$ be a pre-defined feature function. Function s_{shift} is defined as

$$s_{shift}(x, S) = \boldsymbol{w} \cdot \boldsymbol{\phi}(x, S). \quad (1)$$

Since this model is a linear discriminative model, there are well-known algorithms to estimate the parameters of the model.

Usually, such models are trained with each occurrence of words as one instance (word-wise learning). However, we can train our model more effectively with each sentence being one instance (sentence-wise learning). In this section, we describe how to train our model in two different ways and how to apply the model to a sentence classification.

4.1 Word-wise Learning

In this learning method, we train the word-level polarity-shift model with each occurrence of sentiment words being an instance. Training examples are automatically extracted by finding sentiment words in labeled sentences. In the example of Section 3, for instance, both negative words ("distortion" or "problems") and a positive word ("pleased") appear in a positive sentence. We regard "distortion" and "problems", whose polarities are different from that of the sentence, as belonging to the *polarity-shifted* class. On the contrary, we regard "pleased", whose polarity is the same as that of the sentence, as not belonging to *polarity-shifted* class.

We can use the majority voting by those (possibly polarity-shifted) sentiment words. Specifically, we first classify each sentiment word in the sentence according to whether the polarity is shifted or not. Then we use the majority voting to determine the polarity of the sentence. If the first classifier classifies a positive word into the "polarity-shifted" class, we treat the word as a negative one. We expect that the majority voting with polarity-shifting will outperform the simple majority voting without polarity-shifting. We actually use the weighted majority voting, where the polarity-shifting score for each sentiment word is used as the weight of the vote by the word. We expect that the score works as a confidence measure.

We can formulate this method as follows. Here, N and P are respectively defined as the sets of negative sentiment words and positive sentiment words. For instance, $x \in N$ means that x is a negative word. We also write $x \in S$ to express that the word x occurs in S.

First, let us define two scores, $score_p(S)$ and $score_n(S)$, for the input sentence S. The $score_p(S)$ and the $score_n(S)$ respectively represent the number of votes for S being positive and the number of votes for S being negative. If $score_p(S) > score_n(S)$, we regard the sentence S as having the positive polarity, otherwise negative. We suppose that the following relations hold for the scores:

$$score_p(S) = \sum_{x \in P \cap S} -s_{shift}(x, S) + \sum_{x \in N \cap S} s_{shift}(x, S), \quad (2)$$

$$score_n(S) = \sum_{x \in P \cap S} s_{shift}(x, S) + \sum_{x \in N \cap S} -s_{shift}(x, S). \quad (3)$$

When either a polarity-unchanged positive word ($s_{shift}(x, S) \leq 0$) or a polarity-shifted negative word occurs in the sentence S, $score_p(S)$ increases. We can easily obtain the following relation between two scores:

$$score_p(S) = -score_n(S). \quad (4)$$

Since, according to this relation, $score_p(S) > score_n(S)$ is equivalent to $score_p(S) > 0$, we use only $score_p(S)$ for the rest of this paper.

4.2 Sentence-wise Learning

The equation (2) can be rewritten as

$$\begin{aligned} score_p(S) &= \sum_{x \in S} s_{shift}(x, S) I(x) \\ &= \sum_{x \in S} \boldsymbol{w} \cdot \boldsymbol{\phi}(x, S) I(x) \\ &= \boldsymbol{w} \cdot \left\{ \sum_{x \in S} \boldsymbol{\phi}(x, S) I(x) \right\}, \quad (5) \end{aligned}$$

where $I(x)$ is the function defined as follows:

$$I(x) = \begin{cases} +1 & \text{if } x \in N, \\ -1 & \text{if } x \in P, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

This $score_p(S)$ can also be seen as a linear discriminative model and the parameters of the model can be estimated directly (i.e., without carrying out word-wise learning). Each labeled sentence in a corpus can be used as a training instance for the model.

In this method, the model is learned so that the predictive ability for sentence classification is optimized, instead of the predictive ability for polarity-shifting. Therefore, this model can remain indecisive on the classification of word instances that have little contextual evidence about whether polarity-shifting occurs or not. The model can rely more heavily on word instances that have much evidence.

In contrast, the word-wise learning trains the model with all the sentiment words appearing in a corpus. It is assumed here that all the sentiment words have relations with the sentence-level polarity, and that we can always find the evidence of the phenomena that the polarity of a word is different from that of a sentence. Obviously, this assumption is not always correct. As a result, the word-wise learning sometimes puts a large weight on a context word that is irrelevant to the polarity-shifting. This might degrade the performance of sentence classification as well as of polarity-shifting.

4.3 Hybrid Model

Both methods described in Sections 4.1 and 4.2 are to predict the sentence-level polarity only with the word-level polarity. On the other hand, several methods that use another set of features, for example, bag-of-words, n-grams or dependency trees, were proposed for the sentence or document classification tasks. We propose to combine our method with existing methods. We refer to it as *hybrid model*.

In recent work, discriminative models including SVM are often used with many different features. These methods are generally represented as

$$score'_p(X) = \boldsymbol{w}' \cdot \boldsymbol{\phi}'(X), \quad (7)$$

where X indicates the target of classification, for example, a sentence or a document. If $score'_p(X) > 0$, X is classified into the target class. $\boldsymbol{\phi}'(X)$ is a feature function. When the method uses the bag-of-words model, $\boldsymbol{\phi}'$ maps X to a vector with each element corresponding to a word.

Here, we define new score function $score_{comb}(S)$ as a linear combination of $score_p(S)$, the score function of our sentence-wise learning, and $score'_p(S)$, the score function of an existing method. Using this, we can write the function as

$$\begin{aligned} score_{comb}(S) &= \lambda score_p(S) + (1 - \lambda) score'_p(S) \\ &= \lambda \sum_{x \in S} \boldsymbol{w} \cdot \boldsymbol{\phi}(x, S) I(x) + (1 - \lambda) \boldsymbol{w}' \cdot \boldsymbol{\phi}'(S) \\ &= \boldsymbol{w}_{comb} \cdot \\ &\quad \left\langle \lambda \sum_{x \in S} \boldsymbol{\phi}(x, S) I(x),\ (1 - \lambda) \boldsymbol{\phi}'(S) \right\rangle. \quad (8) \end{aligned}$$

Note that $\langle\rangle$ indicates the concatenation of two vectors, $\boldsymbol{w}_{comb}$ is defined as $\langle \boldsymbol{w},\ \boldsymbol{w}' \rangle$ and λ is a parameter which controls the influence of the word-level polarity-shifting model. This model is also a discriminative model and we can estimate the parameters with a variety of algorithms including SVMs. We can incorporate additional information like bag-of-words or dependency trees by $\boldsymbol{\phi}'(S)$.

4.4 Discussions on the Proposed Model

Features such as n-grams or dependency trees can also capture some negations or polarity-shifters. For example, although "satisfy" is positive, the bigram model will learn "not satisfy" as a feature correlated with negative polarity if it appears in the training data. However, the bigram model cannot generalize the learned knowledge to other features such

Table 1: Statistics of the corpus

	customer	movie
# of Labeled Sentences	1,700	10,662
Available	1,436	9,492
# of Sentiment Words	3,276	26,493
Inconsistent Words	1,076	10,674

as "not great" or "not disappoint". On the other hand, our polarity-shifter model learns that the word "not" causes polarity-shifts. Therefore, even if there was no "not disappoint" in training data, our model can determine that "not disappoint" has correlation with positive class, because the dictionary contains "disappoint" as a negative word. For this reason, the polarity-shifting model can be learned even with smaller training data.

What we can obtain from the proposed method is not only a set of polarity-shifters. We can also obtain the weight vector $\boldsymbol{w}$, which indicates the strength of each polarity-shifter and is learned so that the predictive ability of sentence classification is optimized especially in the sentence-wise learning. It is impossible to manually determine such weights for numerous features.

It is also worth noting that all the models proposed in this paper can be represented as a kernel function. For example, the hybrid model can be seen as the following kernel:

$$K_{comb}(S_1, S_2) = \lambda \sum_{x_i \in S_1} \sum_{x_j \in S_2} K((x_i, S_1), (x_j, S_2)) + (1-\lambda) K'(S_1, S_2). \quad (9)$$

Here, K means the kernel function between words and K' means the kernel function between sentences respectively. In addition, $\sum_{x_i} \sum_{x_j} K((x_i, S_1), (x_j, S_2))$ can be seen as an instance of *convolution kernels*, which was proposed by Haussler (1999). Convolution kernels are a general class of kernel functions which are calculated on the basis of kernels between substructures of inputs. Our proposed kernel treats sentences as input, and treats sentiment words as substructures of sentences. We can use high degree polynomial kernels as both K which is a kernel between substructures, i.e. sentiment words, of sentences, and K' which is a kernel between sentences to make the classifiers take into consideration the combination of features.

5 Evaluation

5.1 Datasets

We used two datasets, customer reviews [1] (Hu and Liu, 2004) and movie reviews [2] (Pang and Lee, 2005) to evaluate sentiment classification of sentences. Both of these two datasets are often used for evaluation in sentiment analysis researches. The number of examples and other statistics of the datasets are shown in Table 1.

Our method cannot be applied to sentences which contain no sentiment words. We therefore eliminated such sentences from the datasets. "Available" in Table 1 means the number of examples to which our method can be applied. "Sentiment Words" shows the number of sentiment words that are found in the given sentences. Please remember that sentiment words are defined as those words that are listed in a predefined polarity dictionary in this paper. "Inconsistent Words" shows the number of the words whose polarities conflicted with the polarity of the sentence.

We performed 5-fold cross-validation and used the classification accuracy as the evaluation measure. We extracted sentiment words from General Inquirer (Stone et al., 1996) and constructed a polarity dictionary. After some preprocessing, the dictionary contains 2,084 positive words and 2,685 negative words.

5.2 Experimental Settings

We employed the Max Margin Online Learning Algorithms for parameter estimation of the model (Crammer et al., 2006; McDonald et al., 2007). In preliminary experiments, this algorithm yielded equal or better results compared to SVMs. As the feature representation, $\boldsymbol{\phi}(x, S)$, of polarity-shifting model, we used the local context of three words to the left and right of the target sentiment word. We used the polynomial kernel of degree 2 for polarity-shifting model and the linear kernel for oth-

[1] http://www.cs.uic.edu/~liub/FBS/FBS.html

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/

Table 2: Experimental results of the sentence classification

methods	customer	movie
Baseline	0.638	0.504
BoW	0.790	0.724
2gram	0.809	0.756
3gram	0.800	**0.762**
Simple-Voting	0.716	0.624
Negation Voting	0.733	0.658
Word-wise	0.783	0.699
Sentence-wise	0.806	0.718
Hybrid BoW	0.827	0.748
Hybrid 2gram	**0.840**	0.755
Hybrid 3gram	0.837	0.758
Opt	**0.840**	**0.770**

ers, and feature vectors are normalized to 1. In hybrid models, the feature vectors, $\sum_{x \in S} \phi(x, S) I(x)$ and $\phi'(S)$ are normalized respectively.

5.3 Comparison of the Methods

We compared the following methods:

- **Baseline** classifies all sentences as positive.
- **BoW** uses unigram features. **2gram** uses unigrams and bigrams. **3gram** uses unigrams, bigrams, and 3grams.
- **Simple-Voting** is the most simple majority voting with word-level polarity (Section 3).
- **Negation Voting** proposed by Hu and Liu (2004) is the majority voting that takes negations into account. As negations, we employed *not*, *no*, *yet*, *never*, *none*, *nobody*, *nowhere*, *nothing*, and *neither*, which are taken from (Polanyi and Zaenen, 2004; Kennedy and Inkpen, 2006; Hu and Liu, 2004) (Section 3).
- **Word-wise** was described in Section 4.1.
- **Sentence-wise** was described in Section 4.2.
- **Hybrid BoW, hybrid 2gram, hybrid 3gram** are combinations of sentence-wise model and respectively **BoW**, **2gram** and **3gram** (Section 4.3). We set $\lambda = 0.5$.

Table 2 shows the results of these experiments. Hybrid 3gram, which corresponds to the proposed method, obtained the best accuracy on customer review dataset. However, on movie review dataset, the proposed method did not outperform 3gram. In Section 5.4, we will discuss this result in details. Comparing word-wise to simple-voting, the accuracy increased by about 7 points. This means that the polarity-shifting model can capture the polarity-shifts and it is an important factor for sentiment classification. In addition, we can see the effectiveness of sentence-wise, by comparing it to word-wise in accuracy.

"Opt" in Table 2 shows the results of hybrid models with optimal λ and combination of models. The optimal results of hybrid models achieved the best accuracy on both datasets.

We show some dominating polarity-shifters obtained through learning. We obtained many negations (e.g., no, not, n't, never), modal verbs (e.g., might, would, may), prepositions (e.g., without, despite), comma with a conjunction (e.g., ", but" as in "the case is strong and stylish, but lacks a window"), and idiomatic expressions (e.g., "hard resist" as in "it is hard to resist", and "real snooze").

5.4 Effect of Training Data Size

When we have a large amount of training data, the n-gram classifier can learn well whether each n-gram tends to appear in the positive class or the negative class. However, when we have only a small amount of training data, the n-gram classifier cannot capture such tendency. Therefore the external knowledge, such as word-level polarity, could be more valuable information for classification. Thus it is expected that the sentence-wise model and the hybrid model will outperform n-gram classifier which does not take word-level polarity into account, more largely with few training data.

To verify this conjecture, we conducted experiments by changing the number of the training examples, i.e., the labeled sentences. We evaluated three models: sentence-wise, 3gram model and hybrid 3gram on both customer review and movie review.

Figures 1 and 2 show the results on customer review and movie review respectively. When the size of the training data is small, sentence-wise outper-

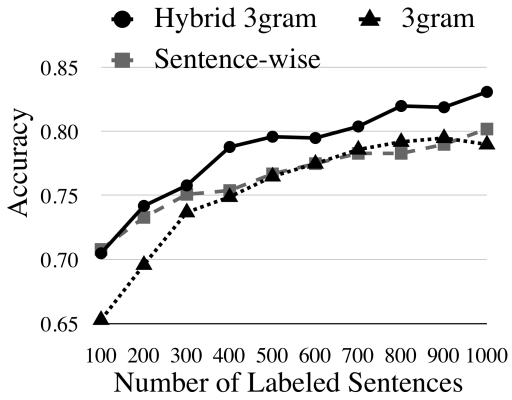


Figure 1: Experimental results on customer review

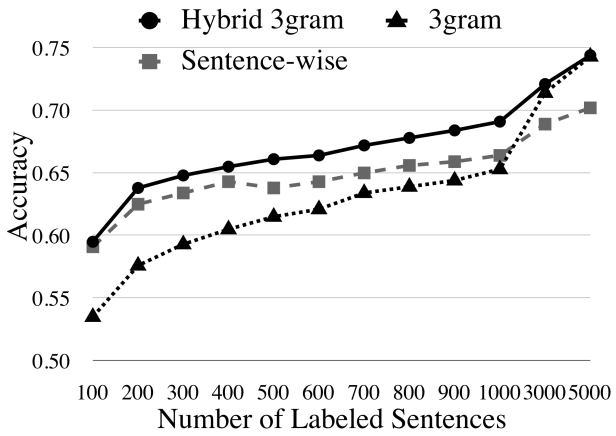


Figure 2: Experimental results on movie review

forms 3gram on both datasets. We can also see that the advantage of sentence-wise becomes smaller as the amount of training data increases, and that the hybrid 3gram model almost always achieved the best accuracy among the three models. Similar behaviour was observed when we ran the same experiments with 2gram or BoW model. From these results, we can conclude that, as we expected above, the word-level polarity is especially effective when we have only a limited amount of training data, and that the hybrid model can combine two models effectively.

6 Conclusion

We proposed a model that captures the polarity-shifting of sentiment words in sentences. We also presented two different learning methods for the model and proposed an augmented hybrid classifier that is based both on the model and on existing classifiers. We evaluated our method and reported that the proposed method almost always improved the accuracy of sentence classification compared with other simpler methods. The improvement was more significant when we have only a limited amount of training data.

For future work, we plan to explore new feature sets appropriate for our model. The feature sets we used for evaluation in this paper are not necessarily optimal and we can expect a better performance by exploring appropriate features. For example, dependency relations between words or appearances of conjunctions will be useful. The position of a word in the given sentence is also an important factor in sentiment analysis (Taboada and Grieve, 2004). Furthermore, we should directly take into account the fact that some words do not affect the polarity of the sentence, though the proposed method tackled this problem indirectly. We cannot avoid this problem to use word-level polarity more effectively. Lastly, since we proposed a method for the sentence-level sentiment prediction, our next step is to extend the method to the document-level sentiment prediction.

Acknowledgement

This research was supported in part by Overseas Advanced Educational Research Practice Support Program by Ministry of Education, Culture, Sports, Science and Technology.

References

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online Passive-Aggressive Algorithms. In *Journal of Machine Learning Research*, Vol.7, Mar, pp.551–585, 2006.

Michael Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-2004)* , pp.841–847, 2004.

David Haussler. Convolution Kernels on Discrete Structures, Technical Report UCS-CRL-99-10, University of California in Santa Cruz, 1999.

Minqing Hu and Bing Liu. Mining Opinion Features in Customer Reviews. In *Proceedings of Nineteeth National Conference on Artificial Intellgience (AAAI-2004)* , pp.755–560, San Jose, USA, July 2004.

Alistair Kennedy and Diana Inkpen. Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. In *Workshop on the Analysis of Formal and Informal Information Exchange during Negotiations (FINEXIN-2005)*, 2005.

Taku Kudo and Yuji Matsumoto. A Boosting Algorithm for Classification of Semi-Structured Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp.301–308, 2004.

Yu Mao and Guy Lebanon. Isotonic Conditional Random Fields and Local Sentiment Flow. In *Proceedings of the Newral Information Processing Systems (NIPS-2006)*, pp.961–968, 2006.

Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment Classification using Word Sub-Sequences and Dependency Sub-Trees. In *Proceedings of the 9th Pacific-Asia International Conference on Knowledge Discovery and Data Mining (PAKDD-2005)*, pp.301–310 , 2005.

Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. Structured Models for Fine-to-Coarse Sentiment Analysis. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-2007)*, pp.432–439, 2007.

Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp.412–418, 2004.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, pp.76–86, 2002.

Bo Pang and Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL-2004)*, pp.271–278, 2004.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, pp.115–124, 2005.

Livia Polanyi and Annie Zaenen. Contextual Valence Shifters. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications (AAAI-EAAT2004)*, 2004.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press, 1996.

Maite Taboada and Jack Grieve. Analyzing Appraisal Automatically. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications (AAAI-EAAT2004)*, pp.158–161, 2004.

Unsupervised Classification of Sentiment and Objectivity in Chinese Text

Taras Zagibalov John Carroll

University of Sussex
Department of Informatics
Brighton BN1 9QH, UK

{T.Zagibalov,J.A.Carroll}@sussex.ac.uk

Abstract

We address the problem of sentiment and objectivity classification of product reviews in Chinese. Our approach is distinctive in that it treats both positive / negative sentiment and subjectivity / objectivity not as distinct classes but rather as a continuum; we argue that this is desirable from the perspective of would-be customers who read the reviews. We use novel unsupervised techniques, including a one-word 'seed' vocabulary and iterative retraining for sentiment processing, and a criterion of 'sentiment density' for determining the extent to which a document is opinionated. The classifier achieves up to 87% F-measure for sentiment polarity detection.

1 Introduction

Automatic classification of sentiment has been a focus of a number of recent research efforts (e.g. (Turney, 2002; Pang et al., 2002; Dave at al., 2003). An important potential application of such work is in business intelligence: brands and company image are valuable property, so organizations want to know how they are viewed by the media (what the 'spin' is on news stories, and editorials), business analysts (as expressed in stock market reports), customers (for example on product review sites) and their own employees. Another important application is to help people find out others' views about products they have purchased (e.g. consumer electronics), services and entertainment (e.g. movies), stocks and shares (from investor bulletin boards), and so on. In the work reported in this paper we focus on product reviews, with the intended users of the processing being would-be customers.

Our approach is based on the insight that positive and negative sentiments are extreme points in a *continuum* of sentiment, and that intermediate points in this continuum are of potential interest. For instance, in one scenario, someone might want to get an idea of the types of things people are saying about a particular product through reading a sample of reviews covering the spectrum from highly positive, through balanced, to highly negative. (We call a review balanced if it is an opinionated text with an undecided or weak sentiment direction). In another scenario, a would-be customer might only be interested in reading balanced reviews, since they often present more reasoned arguments with fewer unsupported claims. Such a person might therefore want to avoid reviews such as Example (1) – written by a Chinese purchaser of a mobile phone (our English gloss).

(1)
软件不行，发送短信时有时对方接收不到；兼容性也不行，有的手机收到的短信是乱码！还有死机现象！拍照效果次！不是循环或自定义式闹铃，每次都要调，太麻烦了！后盖不够严密！原装配件中无座充！

The software is bad, some sent SMS are never received by the addressee; compatibility is also bad, on some mobile phones the received messages are in a scrambled encoding! And sometimes the phone 'dies'! Photos are horrible! It doesn't have a cyclic or pro-

grammable alarm-clock, you have to set it every time, how cumbersome! The back cover does not fit! The original software has many holes!

In a third scenario, someone might decide they would like only to read opinionated, weakly negative reviews such as Example (2), since these often contain good argumentation while still identifying the most salient bad aspects of a product.

(2)
这机子的反应速度超慢的哦，彩信必须要 30KB 以下才能收，也不支持 MP3 铃声，自带铃声也不好听，时不时的还会死机，本来买的时候挺喜欢的，样子挺独特，红色白色搭配的，挺有个性，也不贵，但是用着实在是总出状况，让人头疼

The response time of this mobile is very long, MMS should be less than 30kb only to be downloaded, also it doesn't support MP3 ring tones, (while) the built-in tunes are not good, and from time to time it 'dies', but when I was buying it I really liked it: very original, very nicely matching red and white colours, it has its individuality, also it's not expensive, but when used it always causes trouble, makes one's head ache

The review contains both positive and negative sentiment covering different aspects of the product, and the fact that it contains a balance of views means that it is likely to be useful for a would-be customer. Moving beyond review classification, more advanced tasks such as automatic summarization of reviews (e.g. Feiguina & LaPalme, 2007) might also benefit from techniques which could distinguish more shades of sentiment than just a binary positive / negative distinction.

A second dimension, orthogonal to positive / negative, is opinionated / unopinionated (or equivalently subjective / objective). When shopping for a product, one might be interested in the physical characteristics of the product or what features the product has, rather than opinions about how well these features work or about how well the product as a whole functions. Thus, if one is looking for a review that contains more factual information than opinion, one might be interested in reviews like Example (3).

(3)
总的感觉这台机器还不错，实用的有：开（关）机闹钟 5 个，800 条（500 个人）电话本，阴阳历显示，时间与日期快速转换，WAP 上网，日程表，记事本等。

(My) overall feeling about this mobile is not bad, it features: 5 alarm-clocks that switch the phone on (off), phone book for 800 items (500 people), lunar and solar calendars, fast switching between time and date modes, WAP networking, organizer, notebook and so on.

This review is mostly neutral (unopinionated), but contains information that could be useful to a would-be customer which might not be in a product specification document, e.g. fast switching between different operating modes. Similarly, would-be customers might be interested in retrieving completely unopinionated documents such as technical descriptions and user manuals. Again, as with sentiment classification, we argue that opinionated and unopinionated texts are not easily distinguishable separate sets, but form a continuum. In this continuum, intermediate points are of interest as well as the extremes.

A major obstacle for automatic classification of sentiment and objectivity is lack of training data, which limits the applicability of approaches based on supervised machine learning. With the rapid growth in textual data and the emergence of new domains of knowledge it is virtually impossible to maintain corpora of tagged data that cover all – or even most – areas of interest. The cost of manual tagging also adds to the problem. Reusing the same corpus for training classifiers for new domains is also not effective: several studies report decreased accuracy in cross-domain classification (Engström, 2004; Aue & Gamon, 2005) a similar problem has also been observed in classification of documents created over different time periods (Read, 2005).

In this paper we describe an unsupervised classification technique which is able to build its own sentiment vocabulary starting from a very small seed vocabulary, using iterative retraining to enlarge the vocabulary. In order to avoid problems of domain dependence, the vocabulary is built using text from the same source as the text which is to be classified. In this paper we work with Chinese, but using a very small seed vocabulary may mean that this approach would in principle need very little linguistic adjustment to be applied to a different

language. Written Chinese has some specific features, one of which is the absence of explicitly marked word boundaries, which makes word-based processing problematic. In keeping with our unsupervised, knowledge-poor approach, we do not use any preliminary word segmentation tools or higher level grammatical analysis.

The paper is structured as follows. Section 2 reviews related work in sentiment classification and more generally in unsupervised training of classifiers. Section 3 describes our datasets, and Section 4 the techniques we use for unsupervised classification and iterative retraining. Sections 5 and 6 describe a number of experiments into how well the approaches work, and Section 7 concludes.

2 Related Work

2.1 Sentiment Classification

Most previous work on the problem of categorizing opinionated texts has focused on the binary classification of positive and negative sentiment (Turney, 2002; Pang et al., 2002; Dave at al., 2003). However, Pang & Lee (2005) describe an approach closer to ours in which they determine an author's evaluation with respect to a multi-point scale, similar to the 'five-star' sentiment scale widely used on review sites. However, authors of reviews are inconsistent in assigning fine-grained ratings and quite often star systems are not consistent between critics. This makes their approach very author-dependent. The main differences are that Pang and Lee use discrete classes (although more than two), not a continuum as in our approach, and use supervised machine learning rather than unsupervised techniques. A similar approach was adopted by Hagedorn et al. (2007), applied to news stories: they defined five classes encoding sentiment intensity and trained their classifier on a manually tagged training corpus. They note that world knowledge is necessary for accurate classification in such open-ended domains.

There has also been previous work on determining whether a given text is factual or expresses opinion (Yu & Hatzivassiloglu, 2003; Pang & Lee, 2004); again this work uses a binary distinction, and supervised rather than unsupervised approaches.

Recent work on classification of *terms* with respect to opinion (Esuli & Sebastiani, 2006) uses a three-category system to characterize the opinion-related properties of word meanings, assigning numerical scores to Positive, Negative and Objective categories. The visualization of these scores somewhat resembles our graphs in Section 5, although we use two orthogonal scales rather than three categories; we are also concerned with classification of documents rather than terms.

2.2 Unsupervised Classification

Abney (2002) compares two major kinds of unsupervised approach to classification (co-training and the Yarowsky algorithm). As we do not use multiple classifiers our approach is quite far from co-training. But it is close to the paradigm described by Yarowsky (1995) and Turney (2002) as it also employs self-training based on a relatively small seed data set which is incrementally enlarged with unlabelled samples. But our approach does not use point-wise mutual information. Instead we use relative frequencies of newly found features in a training subcorpus produced by the previous iteration of the classifier. We also use the smallest possible seed vocabulary, containing just a single word; however there are no restrictions regarding the maximum number of items in the seed vocabulary.

3 Data

3.1 Seed Vocabulary

Our approach starts out with a seed vocabulary consisting of a single word, 好 (*good*). This word is tagged as a positive vocabulary item; initially there are no negative items. The choice of word was arbitrary, and other words with strongly positive or negative meaning would also be plausible seeds. Indeed, 好 might not be the best possible seed, as it is relatively ambiguous: in some contexts it means *to like* or acts as the adverbial *very,* and is often used as part of other words (although usually contributing a positive meaning). But since it is one of the most frequent units in the Chinese language, it is likely to occur in a relatively large number of reviews, which is important for the rapid growth of the vocabulary list.

3.2 Test Corpus

Our test corpus is derived from product reviews harvested from the website IT168[1]. All the reviews were tagged by their authors as either positive or negative overall. Most reviews consist of two or three distinct parts: positive opinions, negative opinions, and comments ('other') – although some

[1]http://product.it168.com

reviews have only one part. We removed duplicate reviews automatically using approximate matching, giving a corpus of 29531 reviews of which 23122 are positive (78%) and 6409 are negative (22%). The total number of different products in the corpus is 10631, the number of product categories is 255, and most of the reviewed products are either software products or consumer electronics. Unfortunately, it appears that some users misused the sentiment tagging facility on the website so quite a lot of reviews have incorrect tags. However, the parts of the reviews are much more reliably identified as being positive or negative so we used these as the items of the test corpus. In the experiments described in this paper we used 2317 reviews of mobile phones of which 1158 are negative and 1159 are positive. Thus random choice would have approximately 50% accuracy if all items were tagged either as negative or positive[2].

4 Method

4.1 Sentiment Classification

As discussed in Section 1, we do not carry out any word segmentation or grammatical processing of input documents. We use a very broad notion of words (or phrases) in the Chinese language. The basic units of processing are 'lexical items', each of which is a sequence of one or more Chinese characters excluding punctuation marks (which may actually form part of a word, a whole word or a sequence of words), and `zones', each of which is a sequence of characters delimited by punctuation marks.

Each zone is classified as either positive or negative based whether positive or negative vocabulary items predominate. In more detail, a simple maximum match algorithm is used to find all lexical items (character sequences) in the zone that are in the vocabulary list. As there are two parts of the vocabulary (positive and negative), we correspondingly calculate two scores using Equation (1)[3],

$$S_i = \frac{L_d}{L_{phrase}} S_d N_d \quad (1)$$

where L_d is the length in characters of a matching lexical item, L_{phrase} is the length of the current zone in characters, S_d is the current sentiment score of the matching lexical item (initially 1.0), and N_d is a negation check coefficient. The negation check is a regular expression which determines if the lexical item is preceded by a negation within its enclosing zone. If a negation is found then N_d is set to –1. The check looks for six frequently occurring negations: 不 (bu), 不会 (buhui), 没有 (meiyou), 摆脱 (baituo), 免去 (mianqu), and 避免 (bimian).

The sentiment score of a zone is the sum of sentiment scores of all the items found in it. In fact there are two competing sentiment scores for every zone: one positive (the sum of all scores of items found in the positive part of the vocabulary list) and one negative (the sum of the scores for the items in the negative part). The sentiment direction of a zone is determined from the maximum of the absolute values of the two competing scores for the zone.

This procedure is applied to all zones in a document, classifying each zone as positive, negative, or neither (in cases where there are no positive or negative vocabulary items in the zone). To determine the sentiment direction of the whole document, the classifier computes the difference between the number of positive and negative zones. If the result is greater than zero the document is classified as positive, and vice versa. If the result is zero the document is balanced or neutral for sentiment.

4.2 Iterative Retraining

The task of iterative retraining is to enlarge the initial seed vocabulary (consisting of a single word as discussed in Section 3.1) into a comprehensive vocabulary list of sentiment-bearing lexical items. In each iteration, the current version of the classifier is run on the product review corpus to classify each document, resulting in a training subcorpus of positive and a negative documents. The subcorpus is used to adjust the scores of existing positive and negative vocabulary items and to find new items to be included in the vocabulary.

Each lexical item that occurs at least twice in the corpus is a candidate for inclusion in the vocabulary list. After candidate items are found, the system calculates their relative frequencies in both the positive and negative parts of the current training subcorpus. The system also checks for negation while counting occurrences: if a lexical item is preceded by a negation, its count is reduced by one. This results in negative counts (and thus negative relative frequencies and scores) for those items that

[2]This corpus is publicly available at http://www.informatics.sussex.ac.uk/users/tz21/it168test.zip

[3]In the first iteration, when we have only one item in the vocabulary, negative zones are found by means of the negation check (so *not* + *good* = negative item).

are usually used with negation; for example, 质量太差了(*the quality is far too bad*) is in the **positive part** of the vocabulary with a score of –1.70. This means that the item was found in reviews classified by the system as positive but it was preceded by a negation. If during classification this item is found in a document it will reduce the positive score for that document (as it is in the positive part of the vocabulary), unless the item is preceded by a negation. In this situation the score will be reversed (multiplied by –1), and the positive score will be increased – see Equation (1) above.

For all candidate items we compare their relative frequencies in the positive and negative documents in the subcorpus using Equation (2).

$$difference = \frac{|F_p - F_n|}{(F_p + F_n)/2} \quad (2)$$

If *difference < 1,* then the frequencies are similar and the item does not have enough distinguishing power, so it is not included in the vocabulary. Otherwise the the sentiment score of the item is (re-) calculated – according to Equation (3) for positive items, and analogously for negative items.

$$\frac{F_p}{F_p + F_n} \quad (3)$$

Finally, the adjusted vocabulary list with the new scores is ready for the next iteration.

4.3 Objectivity Classification

Given a sentiment classification for each zone in a document, we compute sentiment density as the proportion of opinionated zones with respect to the total number of zones in the document. Sentiment density measures the proportion of opinionated text in a document, and thus the degree to which the document as a whole is opinionated.

It should be noted that neither sentiment score nor sentiment density are absolute values, but are relative and only valid for comparing one document with other. Thus, a sentiment density of 0.5 does not mean that the review is half opinionated, half not. It means that the review is less opinionated than a review with density 0.9.

5 Experiments

We ran the system on the product review corpus (Section 3.2) for 20 iterations. The results for binary sentiment classification are shown in Table 1. We see increasing F-measure up to iteration 18, after which both precision and recall start to decrease; we therefore use the version of the classifier as it stood after iteration 18[4]. These figures are only indicative of the classification accuracy of the system. Accuracy might be lower for unseen text, although since our approach is unsupervised we could in principle perform further retraining iterations on any sample of new text to tune the vocabulary list to it.

We also computed a (strong) baseline, using as the vocabulary list the NTU Sentiment Dictionary (Ku et al., 2006)[5] which is intended to contain only sentiment-related words and phrases. We assigned each positive and negative vocabulary item a score of 1 or –1 respectively. This setup achieved 87.77 precision and 77.09 recall on the product review corpus.

In Section 1 we argued that sentiment and objectivity should both be considered as continuums, not

Iteration	Precision	Recall	F-measure
1	77.62	28.43	41.62
2	76.15	73.81	74.96
3	81.15	80.07	80.61
4	83.54	82.79	83.16
5	84.66	83.78	84.22
6	85.51	84.77	85.14
7	86.59	85.76	86.17
8	86.78	86.11	86.44
9	87.15	86.32	86.74
10	87.01	86.37	86.69
11	86.9	86.15	86.53
12	87.05	86.41	86.73
13	86.87	86.19	86.53
14	87.35	86.67	87.01
15	87.13	86.45	86.79
16	87.14	86.5	86.82
17	86.8	86.24	86.52
18	87.57	86.89	87.22
19	87.23	86.67	86.95
20	87.18	86.54	86.86

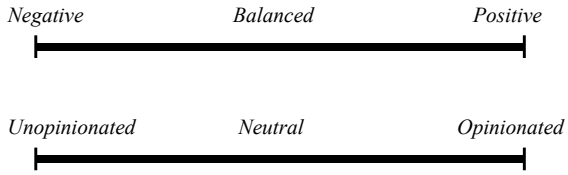
Table 1. Results for binary sentiment classification during iterative retraining.

[4]The size of the sentiment vocabulary after iteration 18 was 22530 (13462 positive and 9068 negative).

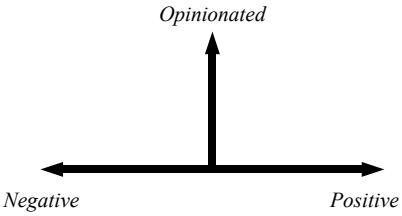
[5]Ku et al. automatically generated the dictionary by enlarging an initial manually created seed vocabulary by consulting two thesauri, including tong2yi4ci2ci2lin2 and the Academia Sinica Bilingual Ontological WordNet 3.

binary distinctions. Section 4.1 describes how our approach compares the number of positive and negative zones for a document and treats the difference as a measure of the 'positivity' or 'negativity' of a review. The document in Example (2), with 12 zones, is assigned a score of –1 (the least negative score possible): the review contains some positive sentiment but the overall sentiment direction of the review is negative. In contrast, Example (1) is identified as a highly negative review, as would be expected, with a score of –8, from 11 zones.

Similarly, with regard to objectivity, the sentiment density of the text in Example (3) is 0.53, which reflects its more factual character compared to Example (1), which has a score of 0.91. We can represent sentiment and objectivity on the following scales:



The scales are orthogonal, so we can combine them into a single coordinate system:



We would expect most product reviews to be placed towards the top of the the coordinate system (i.e. opinionated), and stretch from left to right.

Figure 1 plots the results of sentiment and objectivity classification of the test corpus in this two dimensional coordinate system, where *X* represents sentiment (with scores scaled with respect to the number of zones so that –100 is the most negative possible and +100 the most positive), and *Y* represents sentiment density (0 being unopinionated and 1 being highly opinionated).

Most of the reviews are located in the upper part of the coordinate system, indicating that they have been classified as opinionated, with either positive or negative sentiment direction. Looking at the overall shape of the plot, more opinionated documents tend to have more explicit sentiment direction, while less opinionated texts stay closer to the balanced / neutral region (around *X = 0*).

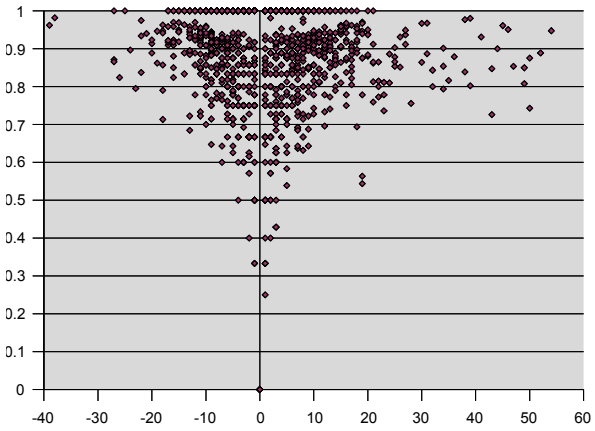


Figure 1. Reviews classified according to sentiment (*X* axis) and degree of opinionation (*Y* axis).

6 Discussion

As can be seen in Figure 1, the classifier managed to map the reviews onto the coordinate system. However, there are very few points in the neutral region, that is, on the same *X = 0* line as balanced but with low sentiment density. By inspection, we know that there are neutral reviews in our data set. We therefore conducted a further experiment to investigate what the problem might be. We took Wikipedia[6] articles written in Chinese on mobile telephony and related issues, as well as several articles about the technology, the market and the history of mobile telecommunications, and split them into small parts (about a paragraph long, to make their size close to the size of the reviews) resulting in a corpus of 115 documents, which we assume to be mostly unopinionated. We processed these documents with the trained classifier and found that they were mapped almost exactly where balanced documents should be (see Figure 2).

Most of these documents have weak sentiment direction (*X = –5* to *+10*), but are classified as relatively opinionated (*Y > 0.5*). The former is to be expected, whereas the latter is not. When investigating the possible reasons for this behavior we noticed that the classifier found not only feature descriptions (like 手感很好 *nice touch*) or expressions which describe attitude (喜欢 *(one) like(s)*), but also product features (for example, 彩信 *MMS* or 电视 *TV*) to be opinionated. This is because the presence of some advanced features such as MMS in mobile phones is often regarded as a positive by

[6]www.wikipedia.org

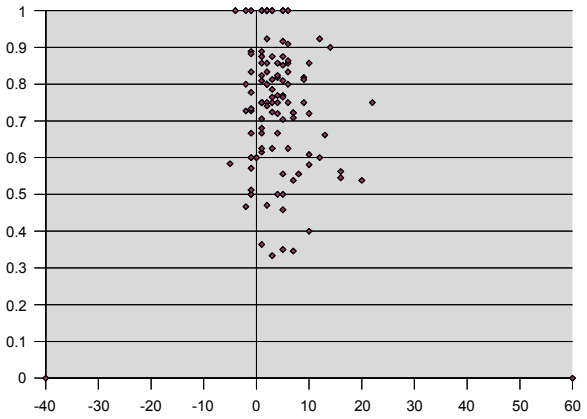


Figure 2. Classification of a sample of articles from Wikipedia.

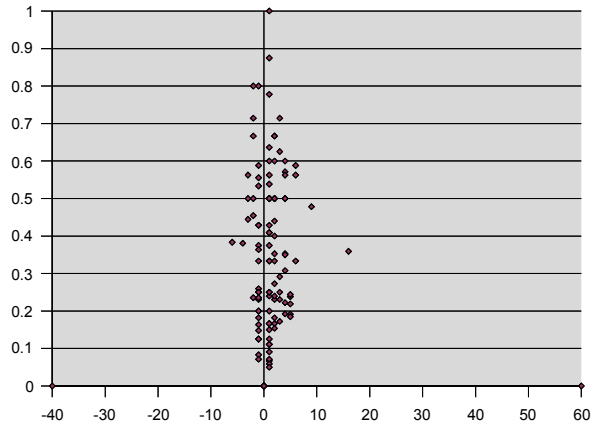


Figure 3. Classification of a sample of articles from Wikipedia, using the NTU Sentiment Dictionary as the vocabulary list.

authors of reviews. In addition, the classifier found words that were used in reviews to describe situations connected with a product and its features: for example, 服务 (*service*) was often used in descriptions of quite unpleasant situations when a user had to turn to a manufacturer's post-sales service for repair or replacement of a malfunctioning phone, and 用户 (*user*) was often used to describe what one can do with some advanced features. Thus the classifier was able to capture some product-specific as well as market-specific sentiment markers, however, it was not able to distinguish the context these generally objective words were used in. This resulted in relatively high sentiment density of neutral texts which contained these words but used in other types of context.

To verify this hypothesis we applied the same processing to our corpus derived from Wikipedia articles, but using as the vocabulary list the NTU Sentiment Dictionary. The results (Figure 3) show that most of the neutral texts are now mapped to the lower part of the opinionation scale ($Y < 0.5$), as expected. Therefore, to successfully distinguish between balanced reviews and neutral documents a classifier should be able to detect when product features are used as sentiment markers and when they are not.

7 Conclusions and Future Work

We have described an approach to classification of documents with respect to sentiment polarity and objectivity, representing both as a continuum, and mapping classified documents onto a coordinate system that also represents the difference between balanced and neutral text. We have presented a novel, unsupervised, iterative retraining procedure for deriving the classifier, starting from the most minimal size seed vocabulary, in conjunction with a simple negation check. We have verified that the approach produces reasonable results. The approach is extremely minimal in terms of language processing technology, giving it good possibilities for porting to different genres, domains and languages.

We also found that the accuracy of the method depends a lot on the seed word chosen. If the word has a relatively low frequency or does not have a definite sentiment-related meaning, the results may be very poor. For example, an antonymous word to 好 (*good*) in Chinese is 坏 (*bad*), but the latter is not a frequent word: the Chinese prefer to say 不好 (*not good*). When this word was used as the seed word, accuracy was little more than 15%. Although the first iteration produced high precision (82%), the size of the extracted subcorpus was only 24 items, resulting in the system being unable to produce a good classifier for the following iterations. Every new iteration produced an even poorer result as each new extracted corpus was of lower accuracy.

On the other hand, it seems that a seed list consisting of several low-frequency one-character words can compensate each other and produce better results by capturing a larger part of the corpus (thus increasing recall). Nevertheless a single word may also produce results even better than those for multiword seed lists. For example, the two-character word 方便 (*comfortable*) as seed reached 91%

accuracy with 90% recall. We can conclude that our method relies on the quality of the seed word. We therefore need to investigate ways of choosing 'lucky' seeds and avoiding 'unlucky' ones.

Future work should also focus on improving classification accuracy: adding a little language-specific knowledge to be able to detect some word boundaries should help; we also plan to experiment with more sophisticated methods of sentiment score calculation. In addition, the notion of 'zone' needs refining and language-specific adjustments (for example, a 'reversed comma' should not be considered to be a zone boundary marker, since this punctuation mark is generally used for the enumeration of related objects).

More experiments are also necessary to determine how the approach works across domains, and further investigation into methods for distinguishing between balanced and neutral text.

Finally, we need to produce a new corpus that would enable us to evaluate the performance of a pre-trained version of the classifier that did not have any prior access to the documents it was classifying: we need the reviews to be tagged not in a binary way as they are now, but in a way that reflects the two continuums we use (sentiment and objectivity).

Acknowledgements

The first author is supported by the Ford Foundation International Fellowships Program.

References

Abney, Steven (2002) Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA. 360–367.

Aue, Anthony & Michael Gamon (2005) Customizing sentiment classifiers to new domains: a case study. In *Proceedings of RANLP-2005*.

Dave, Kushal, Steve Lawrence & David M. Pennock (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the Twelfth International World Wide Web Conference*. 519–528.

Engström, Charlotte (2004) *Topic dependence in sentiment classification*. Unpublished MPhil dissertation, University of Cambridge.

Esuli, Andrea & Fabrizio Sebastiani (2006) SENTIWORDNET: a publicly available lexical resource for opinion mining. In *Proceedings of LREC-06, the 5th Conference on Language Resources and Evaluation*, Genoa, Italy.

Hagedorn, Bennett, Massimiliano Ciaramita & Jordi Atserias (2007) World knowledge in broad-coverage information filtering. In *Proceedings of the 30th ACM SIGIR Conference on Research and Development in Information Retrieval*. 801–802.

Ku, Lun-Wei, Yu-Ting Liang & Hsin-Hsi Chen (2006) Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of the AAAI-2006 Spring Symposium on Computational Approaches to Analyzing Weblogs*, AAAI Technical Report.

Feiguina, Olga & Guy Lapalme (2007) Query-based summarization of customer reviews. In *Proceedings of the 20th Canadian Conference on Artificial Intelligence*, Montreal, Canada. 452–463.

Pang, Bo, Lillian Lee & Shivakumar Vaithyanathan (2002) Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA. 79–86.

Pang, Bo & Lillian Lee (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain. 271–278.

Pang, Bo & Lillian Lee (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, MI. 115–124.

Read, Jonathon (2005) Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the Student Research Workshop at ACL-05*, Ann Arbor, MI.

Turney, Peter D. (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA. 417–424.

Yarowsky, David (1995) Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA. 189–196.

Yu, Hong & Vasileios Hatzivassiloglou (2003) Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan. 129–136.

Using *Roget's Thesaurus* for Fine-grained Emotion Recognition

Saima Aman
School of Information Technology
and Engineering
University of Ottawa, Ottawa, Canada

saman071@site.uottawa.ca

Stan Szpakowicz
School of Information Technology
and Engineering
University of Ottawa, Ottawa, Canada
ICS, Polish Academy of Sciences
Warszawa, Poland

szpak@site.uottawa.ca

Abstract

Recognizing the emotive meaning of text can add another dimension to the understanding of text. We study the task of automatically categorizing sentences in a text into Ekman's six basic emotion categories. We experiment with corpus-based features as well as features derived from two emotion lexicons. One lexicon is automatically built using the classification system of *Roget's Thesaurus*, while the other consists of words extracted from *WordNet-Affect*. Experiments on the data obtained from blogs show that a combination of corpus-based unigram features with emotion-related features provides superior classification performance. We achieve F-measure values that outperform the rule-based baseline method for all emotion classes.

1 Introduction

Recognizing emotions conveyed by a text can provide an insight into the author's intent and sentiment, and can lead to better understanding of the text's content. Emotion recognition in text has recently attracted increased attention of the NLP community (Alm et al., 2005; Liu et al, 2003; Mihalcea and Liu, 2006); it is also one of the tasks at Semeval-2007[1].

Automatic recognition of emotions can be applied in the development of affective interfaces for Computer-Mediated Communication and Human-Computer Interaction. Other areas that can potentially benefit from automatic emotion analysis are personality modeling and profiling (Liu and Maes, 2004), affective interfaces and communication systems (Liu et al, 2003; Neviarouskaya et al., 2007a) consumer feedback analysis, affective tutoring in e-learning systems (Zhang et al., 2006), and text-to-speech synthesis (Alm et al., 2005).

In this study, we address the task of automatically assigning an emotion label to each sentence in the given dataset, indicating the predominant emotion type expressed in the sentence. The possible labels are happiness, sadness, anger, disgust, surprise, fear and no-emotion. Those are Ekman's (1992) six basic emotion categories, and an additional label to account for the absence of a clearly discernible emotion.

We experiment with two types of features for representing text in emotion classification based on machine learning (ML). Features of the first type are a corpus-based unigram representation of text. Features of the second type comprise words that appear in emotion lexicons. One such lexicon consists of words that we automatically extracted from Roget's Thesaurus (1852). We chose words for their semantic similarity to a basic set of terms that represent each emotion category. Another lexicon builds on lists of words for each emotion category, extracted from WordNet-Affect (Strapparava and Valitutti, 2004).

We compare the classification results for groups of features of these two types. We get good results when the features are combined in a series of ML experiments.

[1] Affective Text: Semeval Task at the 4th International Workshop on Semantic Evaluations, 2007, Prague (nlp.cs.swarthmore.edu/semeval/tasks/task14/summary.shtml).

2 Related Work

Research in emotion recognition has focused on discerning emotions along the dimensions of valence (positive / negative) and arousal (calm / excited), and on recognizing distinct emotion categories. We focus on the latter.

Liu et al. (2003) use a real-world commonsense knowledge base to classify sentences into Ekman's (1992) basic emotion categories. They use an ensemble of rule-based affect models to determine the emotional affinity of individual sentences. Neviarouskaya et al. (2007b) also use rules to determine the emotions in sentences in blog posts; their analysis relies on a manually prepared database of words, abbreviations and emoticons labeled with emotion categories.

Since these papers do not report conventional performance metrics such as precision and recall, the effectiveness of their methods cannot be judged empirically. They also disregard statistical learning methods as ineffective for emotion recognition at sentence level. They surmise that the small size of the text input (a sentence) gives insufficient data for statistical analysis, and that statistical methods cannot handle negation. In this paper, we show that ML-based approach with the appropriate combination of features can be applied to distinguishing emotions in text.

Previous work has used lexical resources such as WordNet to automatically acquire emotion-related words for emotion classification experiments. Starting from a set of primary emotion adjectives, Alm et al. (2005) retrieve similar words from *WordNet* utilizing all senses of all words in the synsets that contain the adjectives. They also exploit the synonym and hyponym relations in WordNet to manually find words similar to nominal emotion words. Kamps and Marx (2002) use *WordNet's* synset relations to determine the affective meaning of words. They assign multi-dimensional scores to individual words based on the minimum path length between them and a pair of polar words (such as "good" and "bad") in *WordNet's* structure.

There is also a corpus-driven method of determining the emotional affinity of words: learn probabilistic affective scores of words from large corpora. Mihalcea and Liu (2006) have used this method to assign a happiness factor to words depending on the frequency of their occurrences in happy-labeled blogposts compared to their total frequency in the corpus.

In this paper, we study a new approach to automatically acquiring a wide variety of words that express emotions or emotion-related concepts, using *Roget's Thesaurus* (1852).

3 Emotion-Labeled Data

We have based our study on data collected from blogs. We chose blogs as data source because they are potentially rich in emotion content, and contain good examples of real-world instances of emotions expressed in text. Additionally, text in blogs does not conform to the style of any particular genre per se, and thus offers a variety in writing styles, choice and combination of words, as well as topics. So, the methods learned for discerning emotion using blog data are quite general and therefore applicable to a variety of genres rather than to blogs only.

We retrieved blogs using seed words for all emotion categories. Four human judges manually annotated the blog posts with emotion-related information - every sentence received two judgments. The annotators were required to mark each sentence with one of the eight labels: happiness, sadness, anger, disgust, surprise, fear, mixed-emotion, and no-emotion. The mixed-emotion label was included to handle those sentences that had more than one type of emotion or whose emotion content could not fit into any of the given emotion categories. Sample sentences from the annotated corpus are shown in Fig. 1.

We measured the inter-annotator agreement using Cohen's (1960) kappa. The average pair-wise agreement for different emotion categories ranged from 0.6 to 0.79. In the experiments reported in this paper, we use only those sentences for which there was agreement between both judgments (to form a benchmark for the evaluation of the results of automatic classification). The distribution of emotion categories in the corpus used in our experiments is shown in Table 1.

This was the best summer I have ever experienced.	*(happiness)*
I don't feel like I ever have that kind of privacy where I can talk to God and cry and figure things out.	*(sadness)*
Finally, I got fed up.	*(disgust)*
I can't believe she is finally here!	*(surprise)*

Fig 1. Sample sentences from the corpus

Emotion Class	Number of sentences
Happiness	536
Sadness	173
Anger	179
Disgust	172
Surprise	115
Fear	115
No-emotion	600

Table 1. Distribution of emotion classes

4 A Baseline Approach

We are interested in investigating if emotion in text can be discerned on the basis of its lexical content. A naïve approach to determining the emotional orientation of text is to look for obvious emotion words, such as "happy", "afraid" or "astonished". The presence of one or more words of a particular emotion category in a sentence provides a good premise for interpreting the overall emotion of the sentence. This approach relies on a list of words with prior information about their emotion type, and uses it for sentence-level classification. The obvious advantage is that no training data are required.

For evaluation purposes, we took this approach to develop a baseline system that counts the number of emotion words of each category in a sentence, and then assigns this sentence the category with the largest number of words. Ties were resolved by choosing the emotion label according to an arbitrarily predefined ordering of emotion classes. A sentence containing no emotion word of any type was assigned the *no emotion* category. This system worked with word lists[2] extracted from *WordNet-Affect* (Strapparava and Valitutti, 2004) for six basic emotion categories.

Table 2 shows the precision, recall, and F-measure values for the baseline system. As we have seven classes in our experiments, the class imbalance makes accuracy values less relevant than precision, recall and F-measure. That is why we do not report accuracy values in our results.

The baseline system shows precision values above 50% for all but two classes. This shows the usefulness of this approach. This method, however, fails in the absence of obvious emotion words in the sentence, as indicated by low recall values. Thus, in order to improve recall, we need to increase the ambit of words that are considered emotion-related. An alternative approach is to use ML to learn automatically rules that classify emotion in text.

Class	Precision	Recall	F-Measure
Happiness	0.589	0.390	0.469
Sadness	0.527	0.283	0.368
Anger	0.681	0.262	0.379
Disgust	0.944	0.099	0.179
Surprise	0.318	0.296	0.306
Fear	0.824	0.365	0.506
No-emotion	0.434	0.867	0.579

Table 2. Performance metrics of the baseline system

5 Approach Based on Machine Learning

We study two types of features: corpus-based features and features based on emotion lexicons.

5.1 Corpus-based features

The corpus-based features exploit the statistical characteristics of the data on the basis of the n-gram distribution. In our experiments, we take unigrams (n=1) as features. Unigram models have been previously shown to give good results in sentiment classification tasks (Kennedy and Inkpen, 2006; Pang et al., 2002): unigram representations can capture a variety of lexical combinations and distributions, including those of emotion words. This is particularly important in the case of blogs, whose language is often characterized by frequent use of new words, acronyms (such as "lol"), onomatopoeic words ("haha", "grrr"), and slang, most of which can be captured in a unigram representa-

[2] Emotion words from WordNet-Affect (http://www.cse.unt.edu/~rada/affectivetext/data/WordNetAffectEmotionLists.tar.gz)

Similarity Score	Happiness	Sadness	Anger	Disgust	Surprise	Fear
16	family, home, friends, life, house, loving, partying, bed, pleasure, rest, close, event, lucks, times	crying, lost, wounds, bad, pills, falling, messed, spot, unhappy, pass, black, events, hurts, shocked	pride, fits, stormed, abandoned, bothered, mental, anger, feelings, distractions	shock, disgust, dislike, loathing	plans, catch, expected, early, slid, slipped, earlier, caught, act	nervous, cry, terror, panic, feelings, run, fog, fire, turn, police, faith, battle, war, sounds
14	love, like, feel, pretty, lovely, better, smiling, nice, beautiful, hope, cutest celebrations, warm, desires	ill, bored, feeling, ruin, blow, down, wrong, awful, evil, worry, crushing, bug, death, trouble, dark	hate, burn, upset, dislike, wrong, blood, ill, flaws, bar, defects, bitter, growled, black, slow	hate, pain, horrifying, ill, pills, sad, wear, blood, appalling, end, work, weighed, regrets, bad	left, swing, noticed, worry, times, amazing, stolen, break, interesting, attention	falling, life, stunned, pay, broken, hate, blast, times, hanging, hope, broken, blood, blue
12	gift, treats, adorable, fun, hug, kidding, bigger, great, lighting, won, stars, enjoy, favourite, social, divine	defeat, nasty, boring, ugly, loser, end, victim, sick, hard, serious, aggravating, bothering, burning	lose, throw, offended, hit, power, feel, flaring, pills, broken, life, forgot, ranting	feel, fun, lies, drawn, lose, missed, deprived, lack, sighs, defeat, down, hurt, tears, insulted	realize, pick, wake, sense, jumped, new, late, magic, omen, forget, popped, feel, question, late, throw	fearful, spy, night, upset, feel, chased, hazardous, tomorrow, victim, grim, terrorists, apprehensive

Table 3. Emotion-related words automatically extracted from *Roget's* Thesaurus

tion. Another advantage of a unigram representation is that it does not require any prior knowledge about the data under investigation or the classes to be identified.

For our experiments, we selected all unigrams that occur more than three times in the corpus. This eliminates rare words, as well as foreign-language words and spelling mistakes, which are quite common in blogs. We also excluded words that occur in a list of stopwords - primarily function words that do not generally have emotional connotations. We used the SMART list of stopword[3], with minor modifications. For instance, we removed from the stop list words such as "what" and "why", which may be used in the context of expressing *surprise*.

5.2 Features derived from *Roget's Thesaurus*

We utilized *Roget's Thesaurus* (Jarmasz and Szpakowicz, 2001) to automatically build a lexicon of emotion-related words. The features based on an emotion lexiconrequire prior knowledge about emotion relatedness of words. We extracted this knowledge from the classification system in *Roget's*, which groups related concepts into various levels of a hierarchy. For a detailed account of this classification structure, see Jarmasz and Szpakowicz (2001).

Roget's structure allows the calculation of semantic relatedness between words, based on the path length between the nodes in the structure that represent those words. In case of multiple paths, the shortest path is considered. Jarmasz and Szpakowicz (2004) have introduced a similarity measure derived from path length, which assigns scores ranging from a maximum of 16 to most semantically related words to a minimum of 0 to least related words. They have shown that on semantic similarity tests this measure outperforms several other methods.

To build a lexicon of emotion-related words utilizing *Roget's* structure, we need first to make two decisions: select a primary set of emotion words starting with which we can extract other similar

[3] SMART stopwords list. Used with the SMART information retrieval system at Cornell University (ftp://ftp.cs.cornell.edu/pub/smart/english.stop)

Model	Class	Precision	Recall	F-Measure	Baseline F-Measure
Unigrams	Happiness	**0.840**	0.675	0.740	0.469
	Sadness	**0.619**	0.301	0.405	0.368
	Anger	0.634	0.358	0.457	0.379
	Disgust	**0.772**	0.453	**0.571**	0.179
	Surprise	**0.813**	0.339	0.479	0.306
	Fear	**0.889**	0.487	0.629	0.506
	No-emotion	0.581	0.342	0.431	0.579
Roget's Thesaurus (RT) Features	Happiness	0.772	0.562	0.650	0.469
	Sadness	0.574	0.225	0.324	0.368
	Anger	0.638	0.246	0.355	0.379
	Disgust	0.729	0.297	0.421	0.179
	Surprise	0.778	0.243	0.371	0.306
	Fear	0.857	0.470	0.607	0.506
	No-emotion	0.498	0.258	0.340	0.579
Unigrams + RT Features	Happiness	0.809	**0.705**	**0.754**	0.469
	Sadness	0.577	0.370	**0.451**	0.368
	Anger	0.636	0.419	0.505	0.379
	Disgust	0.686	0.471	0.559	0.179
	Surprise	0.717	0.374	0.491	0.306
	Fear	0.831	0.513	0.634	0.506
	No-emotion	0.586	0.512	0.546	0.579
Unigrams + RT Features + WNA Features	Happiness	0.813	0.698	0.751	0.469
	Sadness	0.605	**0.416**	0.493	0.368
	Anger	**0.650**	**0.436**	**0.522**	0.379
	Disgust	0.672	**0.488**	0.566	0.179
	Surprise	0.723	**0.409**	**0.522**	0.306
	Fear	0.868	**0.513**	**0.645**	0.506
	No-emotion	**0.587**	**0.625**	**0.605**	0.579

* Highest precision, recall, and F-measure values for each class are shown in bold

Table 4 ML Classification Results

words, and choose an appropriate similarity score to serve as cutoff for determining semantic relatedness between words.

The primary set of words that we selected consists of one word for each emotion category, representing the base form of the name of the category: {*happy, sad, anger, disgust, surprise, fear*}.

Experiments performed on Miller and Charles similarity data (1991), reported in Jarmasz and Szpakowicz (2004), have shown that pairs of words with a semantic similarity value of 16 have high similarity, while those with a score of 12 to 14 have intermediate similarity. Therefore, we select the score of 12 as cutoff, and include in the lexicon all words that have similarity scores of 12 or higher with respect to the words in the primary set. This selection of cutoff therefore serves as a form of feature selection. In Table 3, we present sample words from the lexicon with similarity scores of 16, 14, and 12 for each emotion category. These words represent three different levels of relatedness to each emotion category. We are able to identify a large variety of emotion-related words belonging to different parts of speech that go well beyond the stereotypical words associated with different emotions. We particularly note some generic neutral words, such as "feel", "life", and "times" associated with many emotion categories, indicating their conceptual relevance to emotions.

5.3 Features derived from *WordNet-Affect*

WordNet-Affect is an affective lexical resource that assigns a variety of affect-related labels to a subset

of *WordNet* synsets comprising affective concepts. We used lists of words extracted from it for each of the six emotion categories.

6 Experiments and Results

We train classifiers with unigram features for each emotion class using Support Vector Machine (SVM) for predicting the emotion category of the sentences in our corpus. SVM has been shown to be useful for text classification tasks (Joachims, 1998), and has previously given good performance in sentiment classification experiments (Kennedy and Inkpen, 2006; Mullen and Collier, 2004; Pang and Lee, 2004; Pang et al., 2002). In Table 4, we report results from ten-fold cross-validation experiments conducted using the SMO implementation of SVM in Weka (Witten and Frank, 2005). In each experiment, we represent a sentence by a vector indicating the number of times each feature occurs.

In the first experiment, we use only corpus-based unigram features. We obtain high precision values for all emotion classes (as shown in Table 4), and the recall and F-measure values surpass baseline values for all classes except no-emotion. This validates our premise that unigrams can help learn lexical distributions well to accurately predict emotion categories.

Next, we use as features all words in the emotion lexicon acquired from *Roget's Thesaurus* (RT). The F-measure scores beat the baseline for four out of seven classes. When we combine both corpus-based unigrams with RT features, we can increase recall values across all seven classes.

Finally, we add features from WordNet-Affect to the feature set containing corpus unigrams and RT features. This leads to further improvement in overall performance. Combining all features, we achieve highest recall values across all but one class. The resulting F-measure values (ranging from 0.493 to 0.751) surpass the baseline values across all seven classes. This increase was found to be statistically significant (paired t-test, p=0.05).

7 Discussion

We observe that corpus-based features and emotion-related features together contribute to improved performance, better than given by any one type of feature group alone.

Any automatic way of recognizing emotion should inevitably take into account a wide variety of words that are semantically connected to emotions. While some words are obviously affective, many more are only potentially affective. The latter derive their affective property from their associations with emotional concepts. For instance, words like "family", "friends", "home" are not inherently emotional, but because of their well-known semantic association with emotion concepts, their presence in a sentence can be taken as an indicator of emotion expression in the sentence. We can interpret the results as indicators of how much correlation the classifiers can find between the features and the predicted class. Considering our best results using all features, we find that this correlation is highest for the "happy" class, indicated by a precision of 0.813 and recall of 0.698, the highest among all classes. We can therefore conclude that it is easier to discern happiness in text than Ekman's other basic emotions.

8 Conclusions

Working on a corpus of blog sentences annotated with emotion labels, we were able to demonstrate that a combination of corpus-based unigram features and features derived from emotion lexicons can help automatically distinguish basic emotion categories in written text. When used together in an SVM-based learning environment, these features increased recall in all cases and the resulting F-measure values significantly surpassed the baseline scores for all emotion categories.

In addition, we described a method of building an emotion lexicon derived from *Roget's Thesaurus* on the basis of semantic relatedness of words to a set of basic emotion words for each emotion category. The effectiveness of this emotion lexicon was demonstrated in the emotion classification tasks.

References

Cecilia O. Alm, Dan Roth, and Richard Sproat, Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of Joint Conference on HLT/EMNLP*, pages 579-586, Vancouver, Canada, Oct 2005.

M.M. Bradley and P.J. Lang, *Affective norms for English words (ANEW): Instruction manual and affective*

ratings, Technical Report C-1, The Center for Research in Psychophysiology, University of Florida, 1999.

J. Cohen, A coefficient of agreement for nominal scales, *Educational and Psychological Measurement*, 1960, 20 (1): 37–46.

Paul Ekman, An Argument for Basic Emotions, *Cognition and Emotion*, 6, 1992, 169-200.

Mario Jarmasz and Stan Szpakowicz, The Design and Implementation of an Electronic Lexical Knowledge Base. In *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI 2001)*, Ottawa, Canada, June 2001, 325-333.

Mario Jarmasz and Stan Szpakowicz, Roget's Thesaurus and Semantic Similarity. N. Nicolov, K. Bontcheva, G. Angelova, R. Mitkov (eds.) *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, John Benjamins, Amsterdam/Philadelphia, Current Issues in Linguistic Theory, 260, 2004, 111-120.

Thorsten Joachims, Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning (ECML-98)*, pages 137–142.

Jaap Kamps, Maarten Marx, Robert J. Mokken, and Marten de Rijke, Words with attitude, In *Proceedings of the 1st International Conference on Global WordNet*, pages 332-341, Mysore, India, 2002.

Alistair Kennedy and Diana Inkpen, Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 2006, 22(2):110-125.

Hugo Liu, Henry Lieberman, and Ted Selker, A model of textual affect sensing using real-world knowledge. In *Proceedings of the ACM Conference on Intelligent User Interfaces*, 2003, 125–132.

Hugo Liu, and P. Maes, What Would They Think? A Computational Model of Attitudes. In *Proceedings of the ACM International Conference on Intelligent User Interfaces, IUI 2004*, 38-45, ACM Press.

Rada Mihalcea and Hugo Liu, A corpus-based approach to finding happiness, In *Proceedings of the AAAI Spring Symposium on Computational Approaches for Analysis of Weblogs*, Stanford, CA, USA, March 2006.

G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1-28, 1991.

T Mullen and N Collier. Sentiment analysis using support vector machines with diverse information sources. In Dekang Lin and Dekai Wu, editors, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pages 412–418, Barcelona, Spain.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Analysis of affect expressed through the evolving language of online communication. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI-07)*, pages 278-281, Honolulu, Hawaii, USA, 2007a.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka, Narrowing the Social Gap among People involved in Global Dialog: Automatic Emotion Detection in Blog Posts, In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, pages 293-294, Boulder, CO, USA, March 2007b.

A. Ortony, G.L. Clore, and A. Collins, The cognitive structure of emotions. New York: Cambridge University Press, 1988

Bo Pang and Lillian Lee, 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, Barcelona, Spain, pages 271-278.

Bo Pang, Lillian Lee, and S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, 2002, 79–86.

Peter Mark Roget, *Roget's Thesaurus of English Words and Phrases*. Harlow, Essex, England: Longman Group Limited, 1852.

Carlo Strapparava and A Valitutti, WordNet-Affect: an affective extension of WordNet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-2004)*, Lisbon, 2004, 1083-1086.

Ian H. Witten and Eibe Frank. Data Mining: *Practical Machine Learning Tools and Techniques* (2nd ed.), Morgan Kaufmann, San Francisco, 2005. (www.cs.waikato.ac.nz/ml/weka/)

L. Zhang, J. Barnden, R. Hendley, and A. Wallington, Exploitation in Affect Detection in Open-Ended Improvisational Text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text,* 2006, pages 47-54, Sydney, Australia.

Heterogeneous Automatic MT Evaluation Through Non-Parametric Metric Combinations

Jesús Giménez and **Lluís Màrquez**
TALP Research Center, LSI Department
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1–3, E-08034, Barcelona
{jgimenez,lluism}@lsi.upc.edu

Abstract

Combining different metrics into a single measure of quality seems the most direct and natural way to improve over the quality of individual metrics. Recently, several approaches have been suggested (Kulesza and Shieber, 2004; Liu and Gildea, 2007; Albrecht and Hwa, 2007a). Although based on different assumptions, these approaches share the common characteristic of being *parametric*. Their models involve a number of parameters whose weight must be adjusted. As an alternative, in this work, we study the behaviour of *non-parametric* schemes, in which metrics are combined without having to adjust their relative importance. Besides, rather than limiting to the lexical dimension, we work on a wide set of metrics operating at different linguistic levels (e.g., lexical, syntactic and semantic). Experimental results show that non-parametric methods are a valid means of putting different quality dimensions together, thus tracing a possible path towards *heterogeneous* automatic MT evaluation.

1 Introduction

Automatic evaluation metrics have notably accelerated the development cycle of MT systems in the last decade. There exist a large number of metrics based on different similarity criteria. By far, the most widely used metric in recent literature is BLEU (Papineni et al., 2001). Other well-known metrics are WER (Nießen et al., 2000), NIST (Doddington, 2002), GTM (Melamed et al., 2003), ROUGE (Lin and Och, 2004a), METEOR (Banerjee and Lavie, 2005), and TER (Snover et al., 2006), just to name a few. All these metrics take into account information at the lexical level[1], and, therefore, their reliability depends very strongly on the heterogeneity/representativity of the set of reference translations available (Culy and Riehemann, 2003). In order to overcome this limitation several authors have suggested taking advantage of paraphrasing support (Zhou et al., 2006; Kauchak and Barzilay, 2006; Owczarzak et al., 2006). Other authors have tried to exploit information at deeper linguistic levels. For instance, we may find metrics based on full constituent parsing (Liu and Gildea, 2005), and on dependency parsing (Liu and Gildea, 2005; Amigó et al., 2006; Mehay and Brew, 2007; Owczarzak et al., 2007). We may find also metrics at the level of shallow-semantics, e.g., over semantic roles and named entities (Giménez and Màrquez, 2007), and at the properly semantic level, e.g., over discourse representations (Giménez, 2007).

However, none of current metrics provides, in isolation, a *global* measure of quality. Indeed, all metrics focus on *partial* aspects of quality. The main problem of relying on partial metrics is that we may obtain *biased* evaluations, which may lead us to derive *inaccurate* conclusions. For instance, Callison-Burch et al. (2006) and Koehn and Monz (2006) have recently reported several problematic cases related to the automatic evaluation of systems oriented towards maximizing different quality aspects. Corroborating the findings by Culy and Riehemann (2003), they showed that BLEU overrates SMT systems with respect to other types of systems, such

[1] ROUGE and METEOR may consider morphological variations. METEOR may also look up for synonyms in WordNet.

as rule-based, or human-aided. The reason is that SMT systems are likelier to match the sublanguage (e.g., lexical choice and order) represented by the set of reference translations. We argue that, in order to perform more *robust*, i.e., less biased, automatic MT evaluations, different quality dimensions should be jointly taken into account.

A natural solution to this challenge consists in combining the scores conferred by different metrics, ideally covering a *heterogeneous* set of quality aspects. In the last few years, several approaches to metric combination have been suggested (Kulesza and Shieber, 2004; Liu and Gildea, 2007; Albrecht and Hwa, 2007a). In spite of working on a limited set of quality aspects, mostly lexical features, these approaches have provided effective means of combining different metrics into a single measure of quality. All these methods implement a *parametric* combination scheme. Their models involve a number of parameters whose weight must be adjusted (see further details in Section 2).

As an alternative path towards heterogeneous MT evaluation, in this work, we explore the possibility of relying on *non-parametric* combination schemes, in which metrics are combined without having to adjust their relative importance (see Section 3). We have studied their ability to integrate a wide set of metrics operating at different linguistic levels (e.g., lexical, syntactic and semantic) over several evaluation scenarios (see Section 4). We show that non-parametric schemes offer a valid means of putting different quality dimensions together, effectively yielding a significantly improved evaluation quality, both in terms of human likeness and human acceptability. We have also verified that these methods port well across test beds.

2 Related Work

Approaches to metric combination require two important ingredients:

Combination Scheme, i.e., how to combine several metric scores into a single score. As pointed out in Section 1, we distinguish between parametric and non-parametric schemes.

Meta-Evaluation Criterion, i.e., how to evaluate the quality of a metric combination. The two most prominent meta-evaluation criteria are:

- *Human Acceptability:* Metrics are evaluated in terms of their ability to capture the degree of acceptability to humans of automatic translations, i.e., their ability to emulate human assessors. The underlying assumption is that *'good'* translations should be acceptable to human evaluators. Human acceptability is usually measured on the basis of *correlation* between automatic metric scores and human assessments of translation quality[2].
- *Human Likeness:* Metrics are evaluated in terms of their ability to capture the features which distinguish human from automatic translations. The underlying assumption is that *'good'* translations should resemble human translations. Human likeness is usually measured on the basis of *discriminative power* (Lin and Och, 2004b; Amigó et al., 2005).

In the following, we describe the most relevant approaches to metric combination suggested in recent literature. All are parametric, and most of them are based on machine learning techniques. We distinguish between approaches relying on human likeness and approaches relying on human acceptability.

2.1 Approaches based on Human Likeness

The first approach to metric combination based on human likeness was that by Corston-Oliver et al. (2001) who used decision trees to distinguish between human-generated ('good') and machine-generated ('bad') translations. They focused on evaluating only the well-formedness of automatic translations (i.e., subaspects of fluency), obtaining high levels of classification accuracy.

Kulesza and Shieber (2004) extended the approach by Corston-Oliver et al. (2001) to take into account other aspects of quality further than fluency alone. Instead of decision trees, they trained Support Vector Machine (SVM) classifiers. They used features inspired by well-known metrics such as BLEU, NIST, WER, and PER. Metric quality was evaluated both in terms of classification accuracy and correlation with human assessments at the sentence level.

[2] Usually adequacy, fluency, or a combination of the two.

A significant improvement with respect to standard individual metrics was reported.

Gamon et al. (2005) presented a similar approach which, in addition, had the interesting property that the set of human and automatic translations could be independent, i.e., human translations were not required to correspond, as references, to the set of automatic translations.

2.2 Approaches based on Human Acceptability

Quirk (2004) applied supervised machine learning algorithms (e.g., perceptrons, SVMs, decision trees, and linear regression) to approximate human quality judgements instead of distinguishing between human and automatic translations. Similarly to the work by Gamon et al. (2005) their approach does not require human references.

More recently, Albrecht and Hwa (2007a; 2007b) re-examined the SVM classification approach by Kulesza and Shieber (2004) and, inspired by the work of Quirk (2004), suggested a regression-based learning approach to metric combination, with and without human references. The regression model learns a continuous function that approximates human assessments in training examples.

As an alternative to methods based on machine learning techniques, Liu and Gildea (2007) suggested a simpler approach based on linear combinations of metrics. They followed a *Maximum Correlation Training*, i.e., the weight for the contribution of each metric to the overall score was adjusted so as to maximize the level of correlation with human assessments at the sentence level.

As expected, all approaches based on human acceptability have been shown to outperform that of Kulesza and Shieber (2004) in terms of human acceptability. However, no results in terms of human likeness have been provided, thus leaving these comparative studies incomplete.

3 Non-Parametric Combination Schemes

In this section, we provide a brief description of the QARLA framework (Amigó et al., 2005), which is, to our knowledge, the only existing non-parametric approach to metric combination. QARLA is non-parametric because, rather than assigning a weight to the contribution of each metric, the evaluation of a given automatic output a is addressed through a set of independent probabilistic tests (one per metric) in which the goal is to falsify the hypothesis that a is a human reference. The input for QARLA is a set of test cases A (i.e., automatic translations), a set of similarity metrics X, and a set of models R (i.e., human references) for each test case. With such a testbed, QARLA provides the two essential ingredients required for metric combination:

Combination Scheme Metrics are combined inside the QUEEN measure. QUEEN operates under the *unanimity* principle, i.e., the assumption that a 'good' translation must be similar to all human references according to all metrics. $\text{QUEEN}_X(a)$ is defined as the probability, over $R \times R \times R$, that, for every metric in X, the automatic translation a is more similar to a human reference r than two other references, r' and r'', to each other. Formally:

$$\text{QUEEN}_{X,R}(a) = Prob(\forall x \in X : x(a,r) \geq x(r',r''))$$

where $x(a,r)$ stands for the similarity between a and r according to the metric x. Thus, QUEEN allows us to combine different similarity metrics into a single measure, without having to adjust their relative importance. Besides, QUEEN offers two other important advantages which make it really suitable for metric combination: (i) it is *robust* against metric redundancy, i.e., metrics covering similar aspects of quality, and (ii) it is not affected by the scale properties of metrics. The main drawback of the QUEEN measure is that it requires at least three human references, when in most cases only a single reference translation is available.

Meta-evaluation Criterion Metric quality is evaluated using the **KING** measure of human likeness. All human references are assumed to be equally optimal and, while they are likely to be different, the best similarity metric is the one that identifies and uses the features that are common to all human references, grouping them and separating them from automatic translations. Based on QUEEN, KING represents the probability that a human reference

does not receive a lower score than the score attained by *any* automatic translation. Formally:

$$\text{KING}_{A,R}(X) = Prob(\forall a \in A : \text{QUEEN}_{X,R-\{r\}}(r) \geq \text{QUEEN}_{X,R-\{r\}}(a))$$

KING operates, therefore, on the basis of discriminative power. The closest measure to KING is ORANGE (Lin and Och, 2004b), which is, however, not intended for the purpose of metric combination.

Apart from being non-parametric, QARLA exhibits another important feature which differentiates it form other approaches; besides considering the similarity between automatic translations and human references, QARLA also takes into account the distribution of similarities among human references.

However, QARLA is not well suited to port from human likeness to human acceptability. The reason is that QUEEN is, by definition, a very restrictive measure —a 'good' translation must be similar to *all* human references according to *all* metrics. Thus, as the number of metrics increases, it becomes easier to find a metric which does not satisfy the QUEEN assumption. This causes QUEEN values to get close to zero, which turns correlation with human assessments into an impractical meta-evaluation measure.

We have *simulated* a non-parametric scheme based on human acceptability by working on uniformly averaged linear combinations (ULC) of metrics. Our approach is similar to that of Liu and Gildea (2007) except that in our case all the metrics in the combination are equally important[3]. In other words, ULC is indeed a particular case of a parametric scheme, in which the contribution of each metric is not adjusted. Formally:

$$\text{ULC}_X(a, R) = \frac{1}{|X|} \sum_{x \in X} x(a, R)$$

where X is the metric set, and $x(a, R)$ is the similarity between the automatic translation a and the set of references R, for the given test case, according to the metric x. Since correlation with human assessments at the system level is vaguely informative (it is often estimated on very few system samples), we evaluate metric quality in terms of correlation with human assessments at the sentence level (R_{snt}). We use the sum of adequacy and fluency to simulate a global assessment of quality.

[3]That would be assuming that all metrics operate in the same range of values, which is not always the case.

	AE$_{04}$	**CE$_{04}$**	**AE$_{05}$**	**CE$_{05}$**
#human references	5	5	5	4
#system outputs	5	10	7	10
#outputs$_{\text{assessed}}$	5	10	6	5
#sentences	1,353	1,788	1,056	1,082
#sentences$_{\text{assessed}}$	347	447	266	272

Table 1: Description of the test beds

4 Experimental Work

In this section, we study the behavior of the two combination schemes presented in Section 3 in the context of four different evaluation scenarios.

4.1 Experimental Settings

We use the test beds from the 2004 and 2005 NIST MT Evaluation Campaigns (Le and Przybocki, 2005)[4]. Both campaigns include two different translations exercises: Arabic-to-English ('AE') and Chinese-to-English ('CE'). Human assessments of adequacy and fluency are available for a subset of sentences, each evaluated by two different human judges. See, in Table 1, a brief numerical description including the number of human references and system outputs available, as well as the number of sentences per output, and the number of system outputs and sentences per system assessed.

For metric computation, we have used the IQ$_{\text{MT}}$ v2.1, which includes metrics at different linguistic levels (lexical, shallow-syntactic, syntactic, shallow-semantic, and semantic). A detailed description may be found in (Giménez, 2007)[5].

4.2 Evaluating Individual Metrics

Prior to studying the effects of metric combination, we study the isolated behaviour of individual metrics. We have selected a set of metric representatives from each linguistic level. Table 2 shows meta-evaluation results for the test beds described in Section 4.1, according both to human likeness (KING)

[4]`http://www.nist.gov/speech/tests/summaries/2005/mt05.htm`

[5]The IQ$_{\text{MT}}$ Framework may be freely downloaded from `http://www.lsi.upc.edu/~nlp/IQMT`.

Level	Metric	KING				R_{snt}			
		AE_{04}	**CE_{04}**	**AE_{05}**	**CE_{05}**	**AE_{04}**	**CE_{04}**	**AE_{05}**	**CE_{05}**
Lexical	1-WER	0.70	0.51	0.48	0.61	0.53	0.47	0.38	0.47
	1-PER	0.64	0.43	0.45	0.58	0.50	0.51	0.29	0.40
	1-TER	0.73	0.54	0.53	0.66	0.54	0.50	0.38	0.49
	BLEU	0.70	0.49	0.52	0.59	0.50	0.46	0.36	0.39
	NIST	**0.74**	0.53	**0.55**	**0.68**	0.53	**0.55**	0.37	0.46
	GTM.e1	0.67	0.49	0.48	0.61	0.41	0.50	0.26	0.29
	GTM.e2	0.69	0.52	0.51	0.64	0.49	0.54	0.43	0.48
	$ROUGE_L$	0.73	**0.59**	0.49	0.65	**0.58**	**0.60**	0.41	**0.52**
	$ROUGE_W$	**0.75**	**0.62**	**0.54**	**0.68**	**0.59**	**0.57**	**0.48**	**0.54**
	$METEOR_{wnsyn}$	**0.75**	0.56	**0.57**	**0.69**	**0.56**	**0.56**	0.35	0.41
Shallow Syntactic	SP-O_p-*	0.66	0.48	0.49	0.59	0.51	**0.57**	0.38	0.41
	SP-O_c-*	0.65	0.44	0.46	0.59	**0.55**	**0.58**	0.42	0.41
	SP-$NIST_l$	0.73	0.51	**0.55**	**0.66**	0.53	0.54	0.38	0.44
	SP-$NIST_p$	**0.79**	**0.60**	**0.56**	**0.70**	0.46	0.49	0.37	0.39
	SP-$NIST_{iob}$	0.69	0.48	0.49	0.59	0.32	0.36	0.27	0.26
	SP-$NIST_c$	0.60	0.42	0.39	0.52	0.26	0.27	0.16	0.16
Syntactic	DP-HWC_w	0.58	0.40	0.42	0.53	0.41	0.08	0.35	0.40
	DP-HWC_c	0.50	0.32	0.33	0.41	0.41	0.17	0.38	0.32
	DP-HWC_r	0.56	0.40	0.37	0.46	0.42	0.16	0.39	0.43
	DP-O_l-*	0.58	0.48	0.41	0.52	0.52	0.48	0.36	0.37
	DP-O_c-*	0.65	0.45	0.44	0.55	0.49	0.51	0.43	0.41
	DP-O_r-*	0.71	**0.57**	**0.54**	0.64	**0.55**	**0.55**	**0.50**	**0.50**
	CP-O_p-*	0.67	0.47	0.47	0.60	0.53	**0.57**	0.38	0.46
	CP-O_c-*	0.66	0.51	0.49	0.62	**0.57**	**0.59**	**0.45**	0.50
	CP-STM	0.64	0.42	0.43	0.58	0.39	0.13	0.34	0.30
Shallow Semantic	NE-O_e-**	0.65	0.45	0.46	0.57	0.47	0.56	0.32	0.39
	SR-O_r-*	0.48	0.22	0.34	0.41	0.28	0.10	0.32	0.21
	SR-O_{rv}	0.36	0.13	0.24	0.27	0.27	0.12	0.25	0.24
Semantic	DR-O_r-*	0.62	0.47	0.50	0.55	0.47	0.46	0.43	0.37
	DR-O_{rp}-*	0.58	0.42	0.43	0.50	0.37	0.35	0.36	0.26
Optimal Combination		**0.79**	**0.64**	**0.61**	**0.70**	**0.64**	**0.63**	**0.54**	**0.61**

Table 2: Metric Meta-evaluation

and human acceptability (R_{snt}), computed over the subsets of sentences for which human assessments are available.

The first observation is that the two meta-evaluation criteria provide very similar metric quality rankings for a same test bed. This seems to indicate that there is a relationship between the two meta-evaluation criteria employed. We have confirmed this intuition by computing the Pearson correlation coefficient between values in columns 1 to 4 and their counterparts in columns 5 to 8. There exists a high correlation (R = 0.79).

A second observation is that metric quality varies significantly from task to task. This is due to the significant differences among the test beds employed. These are related to three main aspects: language pair, translation domain, and system typology. For instance, notice that most metrics exhibit a lower quality in the case of the 'AE_{05}' test bed. The reason is that, while in the rest of test beds all systems are statistical, the 'AE_{05}' test bed presents the particularity of providing automatic translations produced by *heterogeneous* MT systems (i.e., systems belonging to different paradigms)[6]. The fact that most systems are statistical also explains why, in general, lexical metrics exhibit a higher quality. However, highest levels of quality are not in all cases attained by metrics at the lexical level (see highlighted values). In fact, there is only one metric, '$ROUGE_W$' (based on lexical matching), which is consistently among the top-scoring in all test beds according to both meta-evaluation criteria. The underlying cause is simple: current metrics do not provide a global measure of quality, but account only for partial aspects of it. Apart from evincing the importance of the meta-evaluation process, these results strongly suggest the need for conducting heterogeneous MT evaluations.

[6] Specifically, all systems are statistical except one which is human-aided.

$Opt.K(AE.04)$	$=\{$*SP-NIST$_p$*$\}$
$Opt.K(CE.04)$	$=\{$*ROUGE$_W$*, *SP-NIST$_p$*, *ROUGE$_L$*$\}$
$Opt.K(AE.05)$	$=\{$*METEOR$_{wnsyn}$*, *SP-NIST$_p$*, *DP-O$_r$*-*$\}$
$Opt.K(CE.05)$	$=\{$*SP-NIST$_p$*$\}$
$Opt.R(AE.04)$	$=\{$*ROUGE$_W$*, *ROUGE$_L$*, *CP-O$_c$*-*, *METEOR$_{wnsyn}$*, *DP-O$_r$*-*, *DP-O$_l$*-*, *GTM.e2*, *DR-O$_r$*-*, *CP-STM*$\}$
$Opt.R(CE.04)$	$=\{$*ROUGE$_L$*, *CP-O$_c$*-*, *ROUGE$_W$*, *SP-O$_p$*-*, *METEOR$_{wnsyn}$*, *DP-O$_r$*-*, *GTM.e2*, *1-WER*, *DR-O$_r$*-*$\}$
$Opt.R(AE.05)$	$=\{$*DP-O$_r$*-*, *ROUGE$_W$*$\}$
$Opt.R(CE.05)$	$=\{$*ROUGE$_W$*, *ROUGE$_L$*, *DP-O$_r$*-*, *CP-O$_c$*-*, *1-TER*, *GTM.e2*, *DP-HWC$_r$*, *CP-STM*$\}$

Table 3: Optimal metric sets

4.3 Finding Optimal Metric Combinations

In that respect, we study the applicability of the two combination strategies presented. Optimal metric sets are determined by maximizing over the corresponding meta-evaluation measure (KING or R_{snt}). However, because exploring all possible combinations was not viable, we have used a simple algorithm which performs an *approximate* search. First, individual metrics are ranked according to their quality. Then, following that order, metrics are added to the optimal set only if in doing so the global quality increases. Since no training is required it has not been necessary to keep a held-out portion of the data for test (see Section 4.4 for further discussion).

Optimal metric sets are displayed in Table 3. Inside each set, metrics are sorted in decreasing quality order. The *'Optimal Combination'* line in Table 2 shows the quality attained by these sets, combined under QUEEN in the case of KING optimization, and under ULC in the case of optimizing over R_{snt}. In most cases optimal sets consist of metrics operating at different linguistic levels, mostly at the lexical and syntactic levels. This is coherent with the findings in Section 4.2. Metrics at the semantic level are selected only in two cases, corresponding to the R_{snt} optimization in 'AE$_{04}$' and 'CE$_{04}$' test beds. Also in two cases, corresponding to the KING optimization in 'AE$_{04}$' and 'CE$_{05}$' test beds, it has not been possible to find any metric combination which outperforms the best individual metric. This is not a discouraging result. After all, in these cases, the best metric alone achieves already a very high quality (0.79 and 0.70, respectively). The fact that a single feature suffices to discern between manual and automatic translations indicates that MT systems are easily distinguishable, possibly because of their low quality and/or because they are all based on the same translation paradigm.

4.4 Portability

It can be argued that metric set optimization is itself a training process; each metric would have an associated binary parameter controlling whether it is selected or not. For that reason, in Table 4, we have analyzed the portability of optimal metric sets (i) across test beds and (ii) across combination strategies. As to portability across test beds (i.e., across language pairs and years), the reader must focus on the cells for which the meta-evaluation criterion guiding the metric set optimization matches the criterion used in the evaluation, i.e., the top-left and bottom-right 16-cell quadrangles. The fact that the 4 values in each subcolumn are in a very similar range confirms that optimal metric sets port well across test beds. We have also studied the portability of optimal metric sets across combination strategies. In other words, although QUEEN and ULC are thought to operate on metric combinations respectively optimized on the basis of human likeness and human acceptability, we have studied the effects of applying either measure over metric combinations optimized on the basis of the alternative meta-evaluation criterion. In this case, the reader must compare top-left vs. bottom-left (KING) and top-right vs. bottom-right (R_{snt}) 16-cell quadrangles. It can be clearly seen that optimal metric sets, in general, do not port well across meta-evaluation criteria, particularly from human likeness to human acceptability. However, interestingly, in the case of 'AE$_{05}$' (i.e., heterogeneous systems), the optimal metric set ports well from human acceptability to human likeness. We speculate that system heterogeneity has contributed positively for the sake of robustness.

5 Conclusions

As an alternative to current parametric combination techniques, we have presented two different meth-

Metric Set	KING				R_{snt}			
	$\mathbf{AE_{04}}$	$\mathbf{CE_{04}}$	$\mathbf{AE_{05}}$	$\mathbf{CE_{05}}$	$\mathbf{AE_{04}}$	$\mathbf{CE_{04}}$	$\mathbf{AE_{05}}$	$\mathbf{CE_{05}}$
$\boldsymbol{Opt.K(AE.04)}$	0.79	0.60	0.56	0.70	0.46	0.49	0.37	0.39
$\boldsymbol{Opt.K(CE.04)}$	0.78	0.64	0.57	0.67	0.49	0.51	0.39	0.43
$\boldsymbol{Opt.K(AE.05)}$	0.74	0.63	0.61	0.66	0.48	0.51	0.39	0.42
$\boldsymbol{Opt.K(CE.05)}$	0.79	0.60	0.56	0.70	0.46	0.49	0.37	0.39
$\boldsymbol{Opt.R(AE.04)}$	0.62	0.56	0.52	0.49	0.64	0.61	0.53	0.58
$\boldsymbol{Opt.R(CE.04)}$	0.68	0.59	0.55	0.56	0.63	0.63	0.51	0.57
$\boldsymbol{Opt.R(AE.05)}$	*0.75*	*0.64*	*0.59*	*0.69*	0.62	0.60	0.54	0.57
$\boldsymbol{Opt.R(CE.05)}$	0.64	0.56	0.51	0.52	0.63	0.57	0.53	0.61

Table 4: Portability of combination strategies

ods: a genuine non-parametric method based on human likeness, and a parametric method based human acceptability in which the parameter weights are set equiprobable. We have shown that both strategies may yield a significantly improved quality by combining metrics at different linguistic levels. Besides, we have shown that these methods generalize well across test beds. Thus, a valid path towards heterogeneous automatic MT evaluation has been traced. We strongly believe that future MT evaluation campaigns should benefit from these results specially for the purpose of comparing systems based on different paradigms. These techniques could also be used to build better MT systems by allowing system developers to perform more accurate error analyses and less biased adjustments of system parameters.

As an additional result, we have found that there is a tight relationship between human acceptability and human likeness. This result, coherent with the findings by Amigó et al. (2006), suggests that the two criteria are interchangeable. This would be a point in favour of combination schemes based on human likeness, since human assessments —which are expensive to acquire, subjective and not reusable— are not required. We also interpret this result as an indication that human assessors probably behave in many cases in a discriminative manner. For each test case, assessors would inspect the source sentence and the set of human references trying to identify the features which 'good' translations should comply with, for instance regarding adequacy and fluency. Then, they would evaluate automatic translations roughly according to the number and relevance of the features they share and the ones they do not.

For future work, we plan to study the integration of finer features as well as to conduct a rigorous comparison between parametric and non-parametric combination schemes. This may involve reproducing the works by Kulesza and Shieber (2004) and Albrecht and Hwa (2007a). This would also allow us to evaluate their approaches in terms of both human likeness and human acceptability, and not only on the latter criterion as they have been evaluated so far.

Acknowledgements

This research has been funded by the Spanish Ministry of Education and Science, project OpenMT (TIN2006-15307-C03-02). Our NLP group has been recognized as a Quality Research Group (2005 SGR-00130) by DURSI, the Research Department of the Catalan Government. We are thankful to Enrique Amigó, for his generous help and valuable comments. We are also grateful to the NIST MT Evaluation Campaign organizers, and participants who agreed to share their system outputs and human assessments for the purpose of this research.

References

Joshua Albrecht and Rebecca Hwa. 2007a. A Re-examination of Machine Learning Approaches for Sentence-Level MT Evaluation. In *Proceedings of ACL*, pages 880–887.

Joshua Albrecht and Rebecca Hwa. 2007b. Regression for Sentence-Level MT Evaluation with Pseudo References. In *Proceedings of ACL*, pages 296–303.

Enrique Amigó, Julio Gonzalo, Anselmo Peñas, and Felisa Verdejo. 2005. QARLA: a Framework for the Evaluation of Automatic Sumarization. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics*.

Enrique Amigó, Jesús Giménez, Julio Gonzalo, and Lluís Màrquez. 2006. MT Evaluation: Human-Like vs. Human Acceptable. In *Proceedings of COLING-ACL06*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of EACL*.

Simon Corston-Oliver, Michael Gamon, and Chris Brockett. 2001. A Machine Learning Approach to the Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, pages 140–147.

Christopher Culy and Susanne Z. Riehemann. 2003. The Limits of N-gram Translation Evaluation Metrics. In *Proceedings of MT-SUMMIT IX*, pages 1–8.

George Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proceedings of the 2nd IHLT*.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-Level MT evaluation without reference translations: beyond language modeling. In *Proceedings of EAMT*.

Jesús Giménez and Lluís Màrquez. 2007. Linguistic Features for Automatic Evaluation of Heterogeneous MT Systems. In *Proceedings of the ACL Workshop on Statistical Machine Translation*.

Jesús Giménez. 2007. IQMT v 2.1. Technical Manual. Technical report, TALP Research Center. LSI Department. http://www.lsi.upc.edu/˜nlp/IQMT/-IQMT.v2.1.pdf.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for Automatic Evaluation. In *Proceedings of NLH-NAACL*.

Philipp Koehn and Christof Monz. 2006. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 102–121.

Alex Kulesza and Stuart M. Shieber. 2004. A learning approach to improving sentence-level MT evaluation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Audrey Le and Mark Przybocki. 2005. NIST 2005 machine translation evaluation official results. Technical report, NIST, August.

Chin-Yew Lin and Franz Josef Och. 2004a. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statics. In *Proceedings of ACL*.

Chin-Yew Lin and Franz Josef Och. 2004b. ORANGE: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation. In *Proceedings of COLING*.

Ding Liu and Daniel Gildea. 2005. Syntactic Features for Evaluation of Machine Translation. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Ding Liu and Daniel Gildea. 2007. Source-Language Features and Maximum Correlation Training for Machine Translation Evaluation. In *Proceedings of the 2007 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-07)*.

Dennis Mehay and Chris Brew. 2007. BLEUATRE: Flattening Syntactic Dependencies for MT Evaluation. In *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*.

I. Dan Melamed, Ryan Green, and Joseph P. Turian. 2003. Precision and Recall of Machine Translation. In *Proceedings of HLT/NAACL*.

Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney. 2000. Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of the 2nd LREC*.

Karolina Owczarzak, Declan Groves, Josef Van Genabith, and Andy Way. 2006. Contextual Bitext-Derived Paraphrases in Automatic MT Evaluation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 148–155.

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Dependency-Based Automatic Evaluation for Machine Translation. In *Proceedings of SSST, NAACL-HLT/AMTA Workshop on Syntax and Structure in Statistical Translation*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation, RC22176, IBM. Technical report, IBM T.J. Watson Research Center.

Chris Quirk. 2004. Training a Sentence-Level Machine Translation Confidence Metric. In *Proceedings of LREC*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, , and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of AMTA*, pages 223–231.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating Machine Translation Results with Paraphrase Support. In *Proceedings of EMNLP*.

Paraphrasing depending on Bilingual Context Toward Generalization of Translation Knowledge

Young-Sook Hwang
ETRI
161, Yuseong-gu, Daejeon
305-700, KOREA
yshwang7@etri.re.kr

YoungKil Kim
ETRI
161, Yuseong-gu, Daejeon
305-700, KOREA
kimyk@etri.re.kr

SangKyu Park
ETRI
161, Yuseong-gu, Daejeon
305-700, KOREA
parksk@etri.re.kr

Abstract

This study presents a method to automatically acquire paraphrases using bilingual corpora, which utilizes the bilingual dependency relations obtained by projecting a monolingual dependency parse onto the other language sentence based on statistical alignment techniques. Since the paraphrasing method is capable of clearly disambiguating the sense of an original phrase using the bilingual context of dependency relation, it would be possible to obtain interchangeable paraphrases under a given context. Also, we provide an advanced method to acquire generalized translation knowledge using the extracted paraphrases. We applied the method to acquire the generalized translation knowledge for Korean-English translation. Through experiments with parallel corpora of a Korean and English language pairs, we show that our paraphrasing method effectively extracts paraphrases with high precision, 94.3% and 84.6% respectively for Korean and English, and the translation knowledge extracted from the bilingual corpora could be generalized successfully using the paraphrases with the 12.5% compression ratio.

1 Introduction

Approaches based on bilingual corpora are promising for the automatic acquisition of translation knowledge. Phrase-based SMT(Statistical Machine Translation) models have advanced the state of the art in machine translation by expanding the basic unit of translation from words to phrases, which allows the local reordering of words and translation of multi-word expressions(Chiang, 2007) (Koehn et al., 2003) (Och and Ney, 2004).

However phrase-based SMT techniques suffer from data sparseness problems, that is; unreliable translation probabilities of low frequency phrases and low coverage in that many phrases encountered at run-time are not observed in the training data. An alternative for these problems is to utilize paraphrases. An unknown phrase can be replaced with its paraphrase that is already known. Moreover, we can smooth the phrase translation probability using the class of paraphrases.

On the other hand, EBMT or PBMT systems might translate a given sentence fast and robustly geared by sentence translation patterns or generalized transfer rules. Since it costs too much to construct the translation knowledge, they suffer from the problem of knowledge acquisition bottleneck.

In this study, we present a method of automatically extracting paraphrases from bilingual corpora. Furthermore, we introduce a new method for acquiring the generalized translation knowledge. The translation knowledge is a kind of verb subcategorization pattern composed of bilingual dependency relations. We obtain the generalized translation knowledge by grouping the equivalent constituent phrases. The task of identifying the phrases equivalent to each other is defined as paraphrasing.

Our paraphrasing method utilizes bilingual corpora and alignment techniques in SMT. Unlike pre-

vious approaches which identify paraphrases using a phrase in another language as a pivot without context information (Bannard et al., 2005), or apply the distributional hypothesis to paths in dependency trees for inferring paraphrasing rules from monolingual corpora(Lin et al., 2001), we take the bilingual context of a bilingual dependency relation into account for disambiguating the sense of paraphrases. First, we create a large inventory of bilingual dependency relations and equate the pairs of dependency relations that are aligned with a single dependency relation in the other language as paraphrased dependency relations. Then, we extract the phrases sharing the same head (or modifier) phrase among the paraphrased dependency relations aligned with a unique dependency relation in the other language. We regard them as conceptually equivalent paraphrases. This work is based on the assumption of similar meaning when multiple phrases map onto a single foreign language phrase that is the converse of the assumption made in the word sense disambiguation work(Diab and Resnik, 2002). The two-step paraphrasing method allows us to increase the precision of the paraphrases by constraining the paraphrase candidates under the bilingual contexts of dependency relations.

In order to systematically acquire the generalized translation knowledge, our method includes following steps:

- Derive a bilingually parsed sentence through projecting the source language parse onto the word/phrase aligned target sentence.
- Extract bilingual dependency relations from the bilingual dependency parses.
- Acquire paraphrases by exploiting the extracted bilingual dependency relations.
- Generalize the bilingual dependency relations by substituting the phrases with their paraphrase class.

2 Extracting Translation Patterns

In this section, we introduce a method to acquire translation knowledge like a bilingual dependency pattern using bilingual corpus. The bilingual dependency pattern is defined as an asymmetric binary relationship between a phrase called head and another phrase called modifier which are paired with their corresponding translations in the other language. In order to acquire the bilingual dependency relations, we do bilingual dependency parsing based on the word/phrase alignments and extract bilingual dependency relations by navigating the dependency parse tree.

2.1 Bilingual Dependency Parsing based on Word/Phrase Alignment

Given an input sentence pair, a source language sentence is dependency parsed in a base phrase level and a target language sentence is chunked by a shallow parser. During the dependency parsing and the chunking, each sentence is also segmented into morphemes and we regard a morpheme as a word.

We make word alignments through the learning of IBM models by using the GIZA++ toolkit(Och and Ney, 2000): we learn the translation model toward IBM model 4, initiating translation iterations from IBM model 1 with intermediate HMM model iterations. For improving the word alignment, we use the word-classes that are trained from a monolingual corpus using the srilm toolkit(Stolcke, 2002). Then, we do phrase alignments based on the word alignments, which are consistent with the base phrase boundaries as well as the word alignments as (Hwang et al., 2007) did. A phrase is defined as a word sequence that is covered by a base phrase sequence, not by a single sub-tree in a syntactic parse tree.

After the word and the phrase alignments, we obtain bilingual dependency parses by sharing the dependency relations of a monolingual dependency parser among the aligned phrases. The bilingual dependency parsing is similar to the technique of bilingual parsing in a word level described in (Hwa et al., 2005)(Quirk et al., 2005). Our bilingual parsing in a phrase level has an advantage of being capable of reducing not only the parsing complexity but also the errors caused by structural differences between two languages, such like a Korean and English pairs[1].

For bilingual parsing between Korean and English, we use a Korean dependency parse on the

[1] Since we regard that a phrase in a source language sentence is aligned with a target phrase if at least one word in a source phrase is aligned with the words in a target phrase, we robustly project the source phrases onto the target phrases.

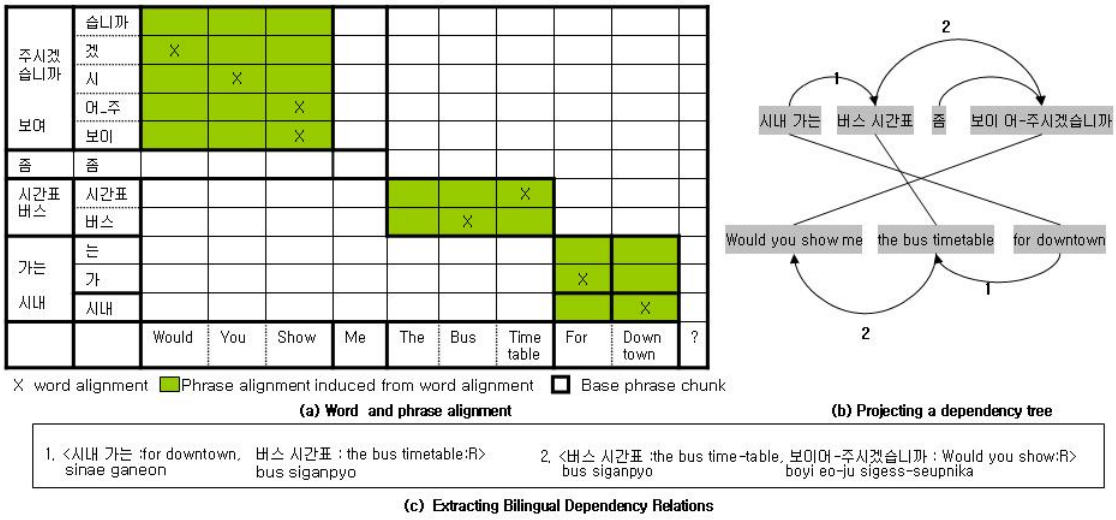


Figure 1: Illustration of Acquiring Bilingual Dependency Relations

source language side as a pivot. Figure 1 shows an illustration of bilingual dependency parsing between Korean and English based on the word/phrase alignments. The dependency structure induced on the target language side is in some sense isomorphic to the structure of the source language.

2.2 Extracting Bilingual Dependency Patterns

Starting from the head phrase of a given source language sentence, we extract bilingual dependency relations by traversing a bilingual dependency parse tree. A dependency relation is a binary relation between a head and modifier phrases. Each phrase is paired with its corresponding translation. For effectively using them during the decoding or the sentence generation, we attach an additional tag for indicating the order(e.g. Reverse or Forward) of target language phrases to the bilingual dependency relation. A dependency pattern refers to the bilingual dependency relation with the phrase order tag.

Figure 1(c) shows some examples of bilingual dependency patterns extracted from the bilingual dependency parse tree in Figure 1(b). In the example, Korean phrase ”sinae ga neun” aligned with the English phrase ”for downtown” modifies the phrase ”bus siganpyo” aligned with the English ”the bus timetable”. Through traversing the dependency parse trees, we acquire the bilingual dependency pattern <sinae ga neun:for downtown, bus siganpyo:the bus timetable;Reverse>.

If we apply the bilingual dependency pattern <sinae ga neun:for downtown, bus siganpyo:the bus timetable;Reverse> for machine translation of a given Korean expression ”sinae ga neun bus siganpyo”, we might generate an English phrase ”the bus timetable for downtown” by reversing the order of English head and modifier phrase corresponding to the Korean phrase ”sinae ga neun bus siganpyo”.

3 Acquisition of Paraphrases

Paraphrasing is based on the assumption that if multiple Korean phrases are equivalent to each other, they can be translated into a single English phrase. But, the reverse is not always true. That is, even though a single phrase in a source language sentence maps onto multiple phrases in a foreign language sentence, the phrases might not be paraphrases. For example, two different Korean phrases, ”gyedan/{stairs,steps}” and ”baldongjak/steps”, might be translated into a single English phrase ”the steps”. But since the meaning of two Korean phrases is not equivalent to each other, the Korean phrases cannot be paraphrases. This implies that the sense of candidate paraphrases should be disambiguated depending on a given context.

For extracting the paraphrases of which sense is disambiguated under a given context, we give a strong constraint on paraphrases with bilingual context evidence of dependency relation denoted as *R(x, y)* :

Figure 2: Illustration of Paraphrasing based on Bilingual Dependency Relations

$$R(e_i, e_j) \equiv R(k_{a_i}, k_{a_j}) \text{ and } R(e_{i'}, e_{j'}) \equiv R(k_{a_{i'}}, k_{a_{j'}}) \Rightarrow R(k_{a_i}, k_{a_j}) \equiv R(k_{a_{i'}}, k_{a_{j'}}) \quad (1)$$

where the relation of $R(e_i, e_j) = R(e_{i'}, e_{j'})$ with the condition of $e_i = e_{i'}$ and $e_j = e_{j'}$.

$$R(e_i, e_j) \equiv R(k_{a_i}, k_{a_j}) \text{ and } R(k_{a_i}, k_{a_j}) \equiv R(k_{a_{i'}}, k_{a_{j'}}) \Rightarrow k_{a_i} \equiv k_{a_{i'}} \text{ iff } k_{a_j} \equiv k_{a_{j'}} \quad (2)$$

For the identification of paraphrases, we equate the different dependency relations aligned with a unique dependency relation in the other language and regard them as a set of paraphrased dependency relations (see eq.(1)). Under the constraint of the paraphrased dependency relations, we again try to acquire paraphrases at a phrase level. That is, we extract the phrases sharing the same head/modifier phrase in paraphrased dependency relations as a phrase paraphrase under a given bilingual dependency context (see eq.(2)).

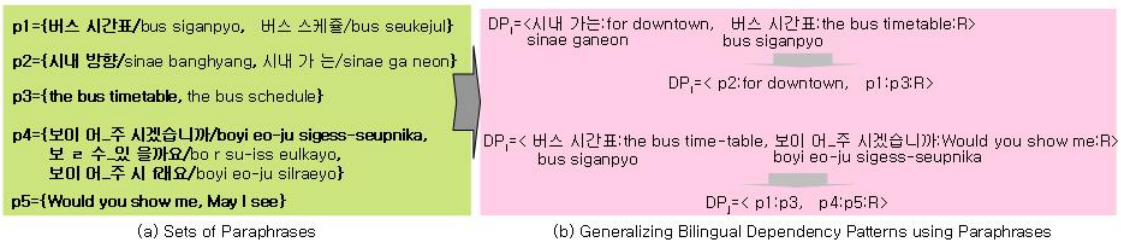
Figure 2 shows some examples of paraphrased dependency relations and paraphrases. In Figure 2 (a), the Korean dependency relations <bus siganpyo, sinae ga neun>,<bus seukejul, sinae ga neun> and <bus seukejul, sinae banghyang> mapped onto the English relation <the bus timetable, for downtown> are the paraphrases. Under the condition of paraphrased dependency relations, the phrases, "bus seukejul" and "bus siganpyo" modified by the same phrase "sinae ga neun" are extracted as paraphrases. In the same way, the set of modifier phrases, p1={"sinae banghyang", "sinae ga neun"} is acquired as a paraphrase set. For English, we obtain the set of paraphrases, p3={"the bus timetable", "the bus schedule"} as we did for Korean.

The induced set of paraphrases can be applied to dependency relations to extend the set through higher inference as in Figure 2(b). We replace a phrase, which is a part of a bilingual dependency relation and a member of a paraphrase set with the representative phrase of the paraphrase set. And we repeatedly apply the paraphrase extraction algorithm to the bilingual dependency relations of which a part is replaced with the previously acquired paraphrase set. Finally, we can acquire new paraphrase sets such as p4 and p5.

4 Generalizing Translation Patterns

The acquired paraphrases can be utilized for various NLP applications. In this work, we focus on making use of the paraphrases to generalize the translation knowledge of bilingual dependency patterns. By generalizing the bilingual dependency patterns, we aim at increasing the coverage of them without any over-generation.

The algorithm for generalizing bilingual dependency patterns is very simple. The main idea is to replace the constituent phrases of a given bilingual dependency pattern with their paraphrase classes. The paraphrase classes are extracted under the condition of a given bilingual context as follows: $< PP(k_m, dp_i) : PP(e_m, dp_i), PP(k_h, dp_i) : PP(e_h, dp_i); Order := Reverse|Forward >$ where the



(a) Sets of Paraphrases (b) Generalizing Bilingual Dependency Patterns using Paraphrases

Figure 3: Illustration of Generalizing Bilingual Dependency Patterns

function, $PP(x, y)$ returns the identifier of the paraphrase set of a given phrase x, which is constrained on a given context $y = dp_i$; k_m and k_h denote a modifier and a head in Korean, respectively and e_m and e_h denote the English phrases.

Figure 3 shows an illustration of generalizing the translation patterns using the previously acquired paraphrase classes. In the pattern dp_i, the English modifier "for downtown" uses the phrase itself because there is no paraphrase class. But, the others are generalized by using their paraphrase classes.

5 Experiments

We used the Basic Travel Expression Corpus (BTEC)(Takezawa et al., 2002), a collection of conversational travel phrases for Korean and English. We used 152,175 sentences in parallel corpus for training and 10,146 sentences for test. The Korean sentences were automatically dependency parsed by in-house dependency parser and the English sentences were chunked by in-house shallow parser.

Through experiments, we investigated the accuracy of the acquired paraphrases, and the compression ratio of the generalized translation patterns compared to the raw translation patterns. Moreover, we show the strength of utilizing bilingual context information in the acquisition of paraphrases with the comparison to the previous approach.

5.1 Accuracy of the Acquired Paraphrases

Through the alignments and bilingual dependency parsing, we extracted 66,664 bilingual dependency relations. 24.15% of Korean phrases and 21.8% of English phrases are paraphrased with more than two phrases under a given bilingual dependency context. The statistics of Korean and English paraphrases based on bilingual dependency relations is shown in Table 1.

Especially, the paraphrasing ratio of the Korean head phrases, 28.63% is higher than that of the English heads,22.6%. Many of the Korean head phrases are verb phrases that reflects the honorific and inflectional characteristics of Korean language. We might expect that the problems caused by various honorific expressions can be resolved with the paraphrases such like {"ga *r geoyeyo*", "ga *gess-seupnida*"}.

For evaluating the accuracy of the acquired paraphrases, we randomly selected 100 sets of paraphrases for Korean and English phrase respectively. Because the accuracy of paraphrases can vary depending on context, we selected the dependency relations that contain a phrase in a paraphrase set from the test set. And we generated the dependency relations by substituting the phrase by the other paraphrases. Accuracy was judged by two native speakers for each language. We measured the percentage of completely interchangeable paraphrases under a given bilingual dependency context.

Table 1 shows the performance of the paraphrases depending on their bilingual context. The accuracy of Korean and English paraphrases are 94.6% and 84.6% respectively. Korean paraphrases are more accurate than English paraphrases. Especially the quality of Korean head paraphrases(97.5%) is very high.

Since we used a simple base-phrase chunker for English, where most base phrases except for noun phrases are composed of single words, most of English phrases aligned to Korean phrases were dependent on the word alignments. Big structural difference between Korean and English made the word alignments more difficult. These alignment results might influence not only the paraphrasing ratio but

	Korean Relation		English Relation	
	Kor-head	Kor-mod	Eng-head	Eng-mod
# of relations	66,664		66,664	
# of uniq relations	59,633		58,187	
	36,157		33,088	
# of uniq phrases	17,867	22,699	13,623	24,000
	6,156		**5,390**	
# of paraphrase set	4,474	2,890	3,425	3,169
	24.15		**21.8**	
Paraphrasing Ratio(%)	28.63	17.7	22.6	19.4
	94.6		**84.6**	
Accuracy(%)	97.5	91.2	86	82.3
Paraphrasing ratio(%) (Bannard et al., 2005)	**44.4**		**37.4**	
accuracy (%) (Bannard et al., 2005)	**71.4**		**76.2**	

Table 1: Statistics of the extracted bilingual dependency relations and paraphrases

also the performance of the paraphrases.

Nevertheless, our paraphrasing method outperformed previous approaches which do not use bilingual dependency context. Because the paraphrasing methods are different, we could not compare them directly. But, we tried to make similar experimental condition on the same BTEC corpus by implementing the previous approach(Bannard et al., 2005). When evaluating the previous approach, the accuracy of (Bannard et al., 2005) was 71.4% and 76.2% for Korean and English paraphrases, respectively. The results show that our paraphrasing method can acquire the paraphrases of higher quality than (Bannard et al., 2005) while the paraphrasing ratio is lower than (Bannard et al., 2005).

5.2 Power of Generalization by Paraphrases

Finally, we investigated how many the extracted bilingual dependency patterns are generalized. Among 66,664 bilingual dependency patterns, 20,968 patterns were generalized into 12,631 unique generalized patterns by applying the extracted paraphrases[2]. As a result, the 66,664 bilingual dependency patterns were compressed into 58,324 generalized patterns with 12.5% compression ratio.

Furthermore, we examined how many bilingual dependency patterns can be generated by the generalized patterns in reverse. When replacing the generalized phrases with all of their paraphrases in both English and Korean sides, 235,640 bilingual translation patterns are generated. These are 3.53 times of the amount of the original translation patterns.

Even we have some errors in the paraphrase sets, these results might contribute to increasing the coverage of the translation knowledge for machine translation.

[2]A paraphrase set is composed of more than two paraphrases

6 Related Work and Discussion

The proposed paraphrasing method can be an extension of the work done by (Bannard et al., 2005). They introduced the method for extracting paraphrases: Using the automatic alignment method from phrase-based SMT, they showed that paraphrases in one language can be identified using a phrase in another language as a pivot. Furthermore, they defined a paraphrase probability to rank the extracted paraphrases and suggested a method to refine it by taking contextual information into account i.e. including simple language model.

Our study for paraphrasing is similar to their work but we take the bilingual dependency context into account for disambiguating the sense of a phrase. Limiting the candidate paraphrases to be the same sense as the original phrase is critical to the performance of paraphrases. Our approach provides the solution to clearly disambiguate the sense of a phrase using bilingual context information. This is the strong point of our approach different from the previous approaches.

Furthermore, in this work, we presented a method to acquire somewhat generalized machine translation knowledge of bilingual dependency patterns. There are few research of the acquisition of translation knowledge such like verb sub-categorization patterns (Fung et al., 2004). (Fung et al., 2004) tried to construct a bilingual semantic network, BiFrameNet to enhance statistical and transfer-

based machine translation systems. They induced the mapping between the English lexical entries in FrameNet to Chinese word senses in HowNet. It takes such an advantage of generalized bilingual frame semantics. But, they have problems of appropriate mapping from lexical entries to word senses and obtaining correct example sentences.

In our approach to acquire the generalized bilingual translation patterns, a bilingual dependency pattern is one of the decomposed bilingual verb subcategorization patterns. It is possible to construct more complicated bilingual verb sub-categorization pattern by applying a kind of unification operation. In that case, we have the advantage of automatically disambiguating the word/phrase senses via the alignment techniques contrary to (Fung et al., 2004).

7 Conclusion

In this paper,we proposed a method to extract paraphrases using bilingual corpora, which utilizes the bilingual dependency relations obtained by projecting a monolingual dependency parse onto the other language sentence based on statistical alignment techniques. The advantage of our paraphrasing method is that it can produce paraphrases of high quality by clearly disambiguating the sense of an original phrase.

Furthermore, we suggested an advanced method to acquire generalized translation knowledge using the extracted paraphrases. With the bilingual dependency patterns generalized by the paraphrases, we aim at reducing the translation ambiguity, but also increasing the coverage of the translation knowledge. The experimental results showed that our generalization method is effective to achieve the goals.

In future, we will utilize the paraphrases based on bilingual dependency relations for increasing the amount of bilingual corpus and for smoothing the phrase probability table in statistical machine translation. Moreover, we plan to apply the acquired translation patterns, which are generalized by paraphrases, to various machine translation systems.

Acknowledgements

This work was supported by the IT R&D program of MIC/IITA, Domain Customization Machine Translation Technology Development for Korean, Chinese, and English.

References

Colin Bannard and Chris Callison Burch. 2005. *Paraphrasing with Bilingual Parallel Corpora*, Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and R. L. Mercer. 1993. *The mathematics of statistical machine translation: Parameter estimation*, Computational Linguistics, 19(2):263-311.

David Chiang. 2007. *Hierarchical phrase-based translation*, Computational Linguistics, 33(2).

M. Diab and P. Resnik. *An Unsupervised Method for Word Sense Tagging Using Parallel Corpora*, Proc. of the 40th Annual Meeting of the Association for Computational Linguistics.

Atsushi Fujita, Kentaro Inui, and Yuji Matsumoto. 2005. *Exploiting Lexical Conceptual Structure for Paraphrase Generation*, Proc. of the 2nd International Joint Conference on Natural Language Processing (IJCNLP).

Pascale Fung and Benfeng Chen 2004 *BiFrameNet: Bilingual Frame Semantics Resource Construction by Cross-lingual Inductio*, Proc. of the 20th International Conference on Computational Linguistics,(COLING 2004),Geneva, Switzerland

Rebeca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas and Okan Kolak. 2005. *Bootstrapping parsers via syntactic projection across parallel texts* , Natural Language Engineering , Vol 11(3), Pages: 311 - 325

Young-Sook Hwang, Andrew Finch and Yutaka Sasaki. 2007. *Improving statistical machine translation using shallow linguistic knowledge*, Computer Speech and Language , Vol. 21(2).

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003 *Statistical Phrase-Based Translation*, Proc. of the Human Language Technology Conference(HLT/NAACL)

D. Lin and P. Pantel 2001. *DIRT-Discovery of Inference Rules from Text*, ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 323-328.

Franz Josef Och and Hermann Ney. 2000. *Improved Statistical Alignment Models* , Proc. of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hongkong, China.

Franz Josef Och and Hermann Ney. 2004. *The alignment template approach to statistical machine translation* , Computational Linguistics, Vol. 30(4), Pages 417-449.

C. Quirk, A. Menezes, and C. Cherry. 2005. *Dependency treelet translation: Syntactically informed phrasal SMT*, Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 271-279.

S. Stolcke 2002 *SRILM - an extensible language modeling toolkit*, Proc. of International Conference of Spoken Language Processing.

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. *Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world*, Proc. of LREC 2002, pp. 147-152, Spain.

A Framework Based on Graphical Models with Logic for Chinese Named Entity Recognition *

Xiaofeng YU **Wai LAM** **Shing-Kit CHAN**

Information Systems Laboratory
Department of Systems Engineering & Engineering Management
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{xfyu,wlam,skchan}@se.cuhk.edu.hk

Abstract

Chinese named entity recognition (NER) has recently been viewed as a *classification* or *sequence labeling* problem, and many approaches have been proposed. However, they tend to address this problem *without* considering linguistic information in Chinese NEs. We propose a new framework based on probabilistic graphical models with first-order logic for Chinese NER. First, we use Conditional Random Fields (CRFs), a standard and theoretically well-founded machine learning method based on undirected graphical models as a base system. Second, we introduce various types of domain knowledge into Markov Logic Networks (MLNs), an effective combination of first-order logic and probabilistic graphical models for validation and error correction of entities. Experimental results show that our framework of probabilistic graphical models with first-order logic significantly outperforms the state-of-the-art models for solving this task.

1 Introduction

Named entity recognition (NER) is the task of identifying and classifying phrases that denote certain types of named entities (NEs), such as person names (PERs), locations (LOCs) and organizations (ORGs) in text documents. It is a well-established task in the NLP and data mining communities and is regarded as crucial technology for many higher-level applications, such as information extraction, question answering, information retrieval and knowledge management. The NER problem has generated much interest and great progress has been made, as evidenced by its inclusion as an understanding task to be evaluated in the Message Understanding Conference (MUC), the Multilingual Entity Task (MET) evaluations, and the Conference on Computational Natural Language Learning (CoNLL).

Compared to European-language NER, Chinese NER seems to be more difficult (Yu *et al.*, 2006). Recent approaches to Chinese NER are a shift away from manually constructed rules or finite state patterns towards machine learning or statistical methods. However, rule-based NER systems lack robustness and portability. Statistical methods often suffer from the problem of data sparsity, and machine learning approaches (e.g., Hidden Markov Models (HMMs) (Bikel *et al.*, 1999; Zhou and Su, 2002), Support Vector Machines (SVMs) (Isozaki and Kazawa, 2002), Maximum Entropy (MaxEnt) (Borthwick, 1999; Chieu and Ng, 2003), Transformation-based Learning (TBL) (Brill, 1995) or variants of them) might be unsatisfactory to learn linguistic information in Chinese NEs. Current state-of-the-art models often view Chinese NER as a *classification* or *sequence labeling* problem *without* considering the linguistic and structural information in Chinese NEs. They assume that entities are independent, however in most cases this assumption does not hold because of the existing relationships among the entities. They seek to locate and identify named entities in text by sequentially classifying tokens (words or characters) as to whether or not they participate in an NE, which is sometimes prone to noise and errors.

In fact, Chinese NEs have distinct linguistic characteristics in their composition and human beings usually use prior knowledge to recognize NEs. For example, about 365 of the highest frequently used surnames cover 99% Chinese surnames (Sun *et al.*, 1995). Some LOCs contain location salient words, while some ORGs contain organization salient words. For the LOC "香港特区/Hong Kong Special Region", "香港/Hong Kong" is the name part and "特区/Special Region" is the salient word. For the ORG "香港特区政府/Hong Kong Special Region Government", "香港/Hong Kong" is the LOC name part, "特区/Special Region" is the LOC salient word and "政府/Government" is the ORG salient word. Some ORGs contain one or more PERs, LOCs and ORGs. A more complex exam-

The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Nos: CUHK 4179/03E and CUHK4193/04E) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050363 and 2050391). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

ple is the nested ORG "北京市海淀区清华大学计算机学院/School of Computer Science, Tsinghua University, Haidian District, Beijing City" which contains two ORGs "清华大学/Tsinghua University" and "计算机学院/School of Computer Science" and two LOCs "北京市/Beijing City" and "海淀区/Haidian District". The two ORGs contain ORG salient words "大学/University" and "学院/School", while the two LOCs contain LOC salient words "市/City" and "区/District" respectively.

Inspired by the above observation, we propose a new framework based on probabilistic graphical models with first-order logic which treats Chinese NER [1] as a *statistical relational learning* (SRL) problem and makes use of domain knowledge. First, we employ Conditional Random Fields (CRFs), a discriminatively trained undirected graphical model which has theoretical justification and has been shown to be an effective approach to segmenting and labeling sequence data, as our base system. We then exploit a variety of domain knowledge into Markov Logic Networks (MLNs), a powerful combination of logic and probability, to validate and correct errors made in the base system. We show how a variety of domain knowledge can be formulated as first-order logic and incorporated into MLNs. We use three Markov chain Monte Carlo (MCMC) algorithms, including Gibbs sampling, Simulated Tempering, as well as MC-SAT, and Maximum a posteriori/Most Probable Explanation (MAP/MPE) algorithm for probabilistic inference in MLNs. Experimental results show that our framework based on graphical models with logic yields substantially better NER results, leading to a relative error reduction of up to 23.75% on the F-measure over state-of-the-art models. McNemar's tests confirm that the improvements we obtained are statistically highly significant.

2 State of the Art

2.1 CRF Model for Chinese NER

Conditional Random Fields (CRFs) (Lafferty *et al.*, 2001) are undirected graphical models trained to maximize the conditional probability of the desired outputs given the corresponding inputs. CRFs have the great flexibility to encode a wide variety of arbitrary, non-independent features and to straightforwardly combine rich domain knowledge. Furthermore, they are discriminatively trained, and are often more accurate than generative models, even with the same features. CRFs have been successfully applied to a number of real-world tasks, including NP chunking (Sha and Pereira, 2003), Chinese word segmentation (Peng *et al.*, 2004), information extraction (Pinto *et al.*, 2003; Peng and McCallum, 2004), named entity identification (McCallum and Li, 2003; Settles, 2004), and many others.

[1] In this paper we only focus on PERs, LOCs and ORGs. Since temporal, numerical and monetary phrases can be well identified with rule-based approaches.

Recently, CRFs have been shown to perform exceptionally well on Chinese NER shared task on the third SIGHAN Chinese language processing bakeoff (SIGHAN-06) (Zhou *et al.*, 2006; Chen *et al.*, 2006b,a). We follow the state-of-the-art CRF models using features that have been shown to be very effective in Chinese NER, namely the current character and its part-of-speech (POS) tag, several characters surrounding (both before and after) the current character and their POS tags, current word and several words surrounding the current word.

We also observe some important issues that significantly influence the performance as follows:

Window size: The primitive window size we use is 5 (2 characters preceding the current character and 2 following the current character). We extend the window size to 7 but find that it slightly hurts. The reason is that CRFs can deal with non-independent features. A larger window size may introduce noisy and irrelevant features.

Feature representation: For character features, we use character identities. For word features, BIES representation (each character is beginning of a word, inside of a word, end of a word, or a single word) is employed.

Labeling scheme: The labeling scheme can be BIO, BIOE or BIOES representation. In BIO representation, each character is tagged as either the beginning of a named entity (B), a character inside a named entity (I), or a character outside a named entity (O). In BIOE, the last character in an entity is labeled as E while in BIOES, single-character entities are labeled as S. In general, BIOES representation is more informative and yields better results than both BIO and BIOE.

2.2 Error Analysis

Even though the CRF model is able to accommodate a large number of well-engineered features which can be easily obtained across languages, some NEs, especially LOCs and ORGs are difficult to identify due to the lack of linguistic or structural characteristics. Since predictions are made token by token, some typical and serious tagging errors are still made, as shown below:

- **ORG is incorrectly tagged as LOC**: In Chinese, many ORGs contain location information. The CRF model only tags the location information (in the ORGs) as LOCs. For example, "唐山理工学院/Tangshan Technical Institute" and "海南省省委/Hainan Provincial Committee " are ORGs and they contain LOCs "唐山/Tangshan" and "海南省/Hainan Province", respectively. "唐山/Tangshan" and "海南省/Hainan Province" are only incorrectly tagged as LOCs. This affects the tagging performance of both ORGs and LOCs.

- **LOC is incorrectly tagged as ORG**: The LOCs "悉尼歌剧院/Sydney Opera" and "北京体育馆/Beijing Gymnasium" are mistakenly tagged as ORGs by the CRF model without taking into account the location salient words "歌剧院/Opera" and "体育馆/Gymnasium".

- **The boundary of entity is tagged incorrectly**: This mistake occurs for all the entities. For example, the PER "汤姆·克鲁斯/Tom Cruise" may be tagged as a PER "汤姆/Tom"; the LOC "不来梅/Bremen" may be tagged as a LOC "来梅/Laimei", which is a meaningless word; the ORG "华为公司/Huawei Corporation" may be tagged as an ORG "华为/Huawei". The reasons for these errors are both complicated and varied. However, some of them are related to linguistic knowledge.

- **Common nouns are incorrectly tagged as entities**: For example, the two common nouns "现代数学/Modern Mathematics" and "格兰士微波炉/Galanz Microwave Oven" may be improperly tagged as a LOC and an ORG. Some tagging errors could be easily rectified. Take the erroneous ORG "市委组织，/City Committee Organizes," for example, intuitively it is not an ORG since an entity cannot span any punctuation.

3 Our Proposed Framework

3.1 Overview

We propose a framework based on probabilistic graphical models with first-order logic for Chinese NER. As shown in Figure 1, the framework is composed of three main components. The CRF model is used as a base model. Then we incorporate domain knowledge that can be well formulated into first-order logic to extract entity candidates from CRF results. Finally, the Markov Logic Network (MLN), an undirected graphical model for *statistical relational learning*, is used to validate and correct the errors made in the base model. We begin by briefly reviewing the necessary background of MLNs, including weight learning and inference.

3.2 Markov Logic Networks

A Markov Network (also known as Markov Random Field) is a model for the joint distribution of a set of variables (Pearl, 1988). It is composed of an undirected graph $G = (V, E)$ and a set of real-valued potential functions ϕ_k. A First-Order Knowledge Base (KB) (Genesereth and Nislsson, 1987) is a set of sentences or formulas in first-order logic.

A Markov Logic Network (MLN) (Richardson and Domingos, 2006) is a KB with a weight attached to each formula (or clause). Together with a set of constants representing objects in the domain, it species a ground Markov Network containing one feature for each possible grounding of a first-order formula F_i in the KB, with the corresponding weight w_i. The basic idea in MLNs is that: when a world violates one formula in the KB it is less probable, but not impossible. The fewer formulas a world violates, the more probable it is. The weights associated with the formulas in an MLN jointly determine the probabilities of those formulas (and vice versa) via a *log-linear model*. An MLN is a statistical relational model that defines a probability distribution over Herbrand interpretations (possible worlds), and can

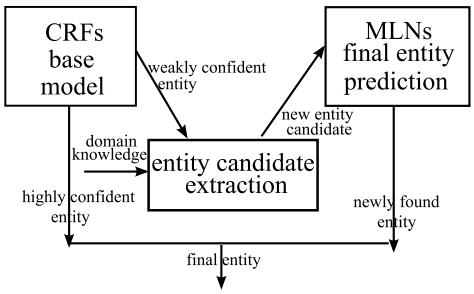


Figure 1: Framework Overview

be thought of as a *template* for constructing Markov Networks. Given different sets of constants, it will produce different networks. These networks will have certain regularities in structure and parameter given by the MLN and they are called ground Markov Networks. Suppose `Peter(A)`, `Smith(B)` and `IBM(X)` are 3 constants, a KB and generated features are listed in Table 1. The formula `Employ(x,y)⇒Person(x),Company(y)` means `x` is employed by `y` and `Colleague(x,y)⇒Employ(x,z)∧Employ(y,z)` means `x` and `y` are colleagues if they are employed by the same company. Figure 2 shows the graph of the ground Markov network defined by the formulas in Table 1 and the 3 constants `Peter(A)`, `Smith(B)` and `IBM(X)`. The probability distribution over possible worlds x specified by the ground Markov Network $M_{L,C}$ is given by

$$P(X = x) = \frac{1}{Z}\exp(\sum w_i n_i(x)) = \frac{1}{Z}\prod \phi_i\left(x_{\{i\}}\right)^{n_i(x)} \quad (1)$$

where $n_i(x)$ is the number of true groundings of F_i in x, $x_{\{i\}}$ is the true value of the atoms appearing in F_i, and $\phi_i\left(x_{\{i\}}\right) = e^{w_i}$.

In the case of Chinese NER, a named entity can be connected to another named entity for instance, because they share the same location salient word. Thus in an undirected graph, two node types exist, the LOC nodes and the location salient word nodes. The links (edges) indicate the relation (LOCs contain location salient words) between them. This representation can be well expressed by MLNs.

However, one problem concerning relational data is, how to extract useful relations for Chinese NER. There are many kinds of relations between NEs, some relations are critical to the NER problem while others not. Another problem that we address is whether these relations can be formulated in first-order logic and combined in MLNs. In Section 3.3, we exploit domain knowledge. We will show how these knowledge can capture essential characteristics of Chinese NEs and can be well and concisely formulated in first-order logic in Section 3.4.

Table 1: Example of a KB and Generated Features

Fist-Order Logic (KB)	Generated Features
∀ x,y Employ(x,y)⇒Person(x),Company(y)	Employ(Peter,IBM)⇒Person(Peter),Company(IBM) Employ(Smith,IBM)⇒Person(Smith),Company(IBM)
∀ x,y,z Colleague(x,y)⇒ Employ(x,z)∧Employ(y,z)	Colleague(Peter,Smith)⇒ Employ(Peter,IBM) ∧Employ(Smith,IBM)

3.2.1 Learning Weights

Given a relational database, MLN weights can in principle be learned generatively by maximizing the likelihood of this database on the closed world assumption. The gradient of the log-likelihood with respect to the weights is

$$\frac{\partial}{\partial w_i}\log P_w(X=x) = n_i(x) - \sum P_w(X=x')n_i(x') \tag{2}$$

where the sum is over all possible databases x', and $P_w(X = x')$ is $P(X = x')$ computed using the current weight vector $w = (w_1, ..., w_i, ...)$. Unfortunately, computing these expectations can be very expensive. Instead, we can maximize the *pseudo-log-likelihood* of the data more efficiently. If x is a possible database and x_l is the *l*th ground atom's truth value, the *pseudo-log-likelihood* of x given weights w is

$$\log P^*_w(X=x) = \sum_{l=1}^{n} \log P_w(X_{l=x_l} \mid MB_x(X_l)) \tag{3}$$

where $MB_x(X_l)$ is the state of X_l's *Markov blanket* [2] in the data. Computing Equation 3 and its gradient does not require inference over the model, and is therefore much faster. We can optimize the *pseudo-log-likelihood* using the limited-memory BFGS algorithm (Liu and Nocedal, 1989).

3.2.2 Inference

If F_1 and F_2 are two formulas in first-order logic, C is a finite set of constants including any constants that appear in F_1 or F_2, and L is an MLN, then

$$\begin{aligned} P(F_1 \mid F_2, L, C) &= P(F_1 \mid F_2, M_{L,C}) \\ &= \frac{P(F_1 \wedge F_2 \mid M_{L,C})}{P(F_2 \mid M_{L,C})} \\ &= \frac{\sum_{x \in \chi_{F_1} \cap \chi_{F_2}} P(X=x \mid M_{L,C})}{\sum_{x \in \chi_{F_2}} P(X=x \mid M_{L,C})} \end{aligned} \tag{4}$$

where χ_{F_i} is the set of worlds where F_i holds, and $P(x \mid M_{L,C})$ is given by Equation 1. The question of whether a knowledge base entails a formula F in first-order logic is the question of whether $P(F \mid L_{\mathrm{KB}}, C_{\mathrm{KB},F}) = 1$, where L_{KB} is the MLN obtained by assigning infinite weight to all the formulas in KB, and $C_{\mathrm{KB},F}$ is the set of all constants appearing in KB or F.

[2] The Markov blanket of a node is the minimal set of nodes that renders it independent of the remaining network; in a MLN, this is simply the node's neighbors in the graph.

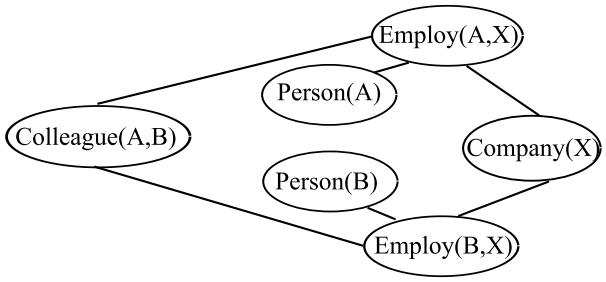


Figure 2: A Ground Markov network defined by the formulas in Table 1 and the constants Peter(A), Smith(B) and IBM(X).

A large number of efficient inference techniques are applicable to MLNs. The most widely used approximate solution to probabilistic inference in MLNs is Markov chain Monte Carlo (MCMC) (Gilks *et al.*, 1996). In this framework, the Gibbs sampling algorithm is to generate an instance from the distribution of each variable in turn, conditional on the current values of the other variables. The key to the Gibbs sampler is that one only considers univariate conditional distributions-the distribution when all of the random variables but one are assigned fixed values. One way to speed up Gibbs sampling is by Simulated Tempering (Marinari and Parisi, 1992), which performs simulation in a *generalized ensemble*, and can rapidly achieve an equilibrium state. Poon and Domingos (2006) proposed MC-SAT, an inference algorithm that combines ideas from MCMC and satisfiability. MC-SAT works well and is guaranteed to be sound, even when deterministic or near-deterministic dependencies are present in real-world reasoning.

Besides MCMC framework, maximum a posteriori (MAP) inference can be carried out using a weighted satisfiability solver like MaxWalkSAT. It is closely related to maximum likelihood (ML), but employs an augmented optimization objective which incorporates a prior distribution over the quantity one wants to estimate. MAP estimation can therefore be seen as a regularization of ML estimation.

3.3 Domain Knowledge

We incorporate various kinds of domain knowledge via MLNs to predict the newly extracted NE candidates from

CRF hypotheses. We extract 165 location salient words and 843 organization salient words from Wikipedia[3] and the LDC Chinese-English bi-directional NE lists compiled from Xinhua News database, as shown in Table 2. We also make a punctuation list which contains 18 items and some stopwords which Chinese NEs cannot contain. The stopwords are mainly conjunctions, auxiliary and functional words. We extract new NE candidates from the CRF results according to the following consideration:

- Definitely, if a chunk (a series of continuous characters) occurs in the training data as a PER or a LOC or an ORG, then this chunk should be a PER or a LOC or an ORG in the testing data. In general, a unique string is defined as a PER, it cannot be a LOC somewhere else.
- Obviously, if a tagged entity ends with a location salient word, it is a LOC. If a tagged entity ends with an organization salient word, it is an ORG.
- If a tagged entity is close to a subsequent location salient word, probably they should be combined together as a LOC. The closer they are, the more likely that they should be combined.
- If a series of consecutive tagged entities are close to a subsequent organization salient word, they should probably be combined together as an ORG because an ORG may contain multiple PERs, LOCs and ORGs.
- Similarly, if there exists a series of consecutive tagged entities and the last one is tagged as an ORG, it is likely that all of them should be combined as an ORG.
- Entity length restriction: all kinds of tagged entities cannot exceed 25 Chinese characters.
- Stopword restriction: intuitively, all tagged entities cannot comprise any stopword.
- Punctuation restriction: in general, all tagged entities cannot span any punctuation.
- Since all NEs are proper nouns, the tagged entities should end with noun words.
- The CRF model tags each token (Chinese character) with a conditional probability. A low probability implies a low-confidence prediction. For a chunk with low conditional probabilities, all the above assumptions are adopted (The marginal probabilities are normalized, and probabilities lower than the user-defined threshold are regarded as low conditional probabilities).

All the above domain knowledge can be formulated as first-order logic to construct the structure of MLNs. And all the extracted chunks are accepted as new NE candidates (or common nouns). We train an MLN to recognize them.

[3]http://en.wikipedia.org/wiki/.

Table 2: Domain Knowledge for Chinese NER

Location Salient Word	Organization Salient Word
自治区/Municipality	百货公司/Department Store
火车站/Railway Station	理工学院/Technical Institute
宾馆/Hotel	旅行社/Travel Agency
公园/Park	出版社/Press
高原/Plateau	人事部/Personnel Department
省/Province	银行/Bank
镇/Town	大学/University
市/City	市委/City Committee
Stopword	**Punctuation**
仍然/still	。
但是/but	？
非常/very	，
的/of	；
等/and so on	：
那/that	！

3.4 First-Order Logic Representation

We declared 14 *predicates* (`person(candidate)`, `location(candidate)`, `organization(candidate)`, `endwith(candidate, salientword)`, `closeto(candidate, salientword)`, `containstopword(candidate)`, `containpunctuation(candidate)`, etc) and specified 15 first-order formulas (See Table 3 for some examples) according to the domain knowledge described in Section 3.3. For example, we used `person(candidate)` to specify whether a candidate is a PER. *Formulas* are recursively constructed from atomic formulas using logical connectives and quantifiers. They are constructed using four types of symbols: *constants*, *variables*, *functions*, and *predicates*. *Constant* symbols represent objects in the domain of interest (e.g., "北京/Beijing" and "上海/Shanghai" are LOCs). *Variable* symbols (e.g., `r` and `p`) range over the objects in the domain. To reduce the size of ground Markov Network, variables and constants are *typed*; for example, the variable `r` may range over candidates, and the constant "北京/Beijing" may represent a LOC. *Function* symbols represent mappings from tuples of objects to objects. *Predicate* symbols represent relations among objects (e.g., `person`) in the domain or attributes of objects (e.g., `endwith`). A *ground atom* is an atomic formula all of whose arguments are ground terms (terms containing no variables). For example, the ground atom `location(北京市)` conveys that "北京市/Beijing City" is a LOC.

For example in Table 3, "乌市/Wu City" is mis-tagged as an ORG by the CRF model, but it contains the location salient word "市/City". So it is extracted as a new entity candidate, and the corresponding formula `endwith(r, p)`$\wedge$`locsalientword(p)`$\Rightarrow$`location(r)` means if `r` ends with a location salient word `p`, then it is a LOC. Besides the formulas listed in Table 3, we also specified logic such as `person(p)`$\Rightarrow$`!(location(p) v organization(p))`, which means a candidate `p` can

Table 3: Examples of NE Candidates and First-Order Formulas

Mis-tagged NEs	New NE Candidates	First-Order Logic
希拉里[common noun]	希拉里	occurperson(p)⇒person(p)
凡尔赛[PER]	凡尔赛	occurlocation(p)⇒location(p)
一汽集团[common noun]	一汽集团	occurorganization(p)⇒organization(p)
乌市[ORG]	乌市	endwith(r,p)∧locsalientword(p)⇒location(r)
英政府[LOC]	英政府	endwith(r,p)∧orgsalientword(p)⇒organization(r)
北海[LOC]花园	北海花园	closeto(r,p)∧locsalientword(p)⇒location(r)
瑞士[LOC]联邦	瑞士联邦	closeto(r,p)∧orgsalientword(p)⇒organization(r)
市区的酒店[LOC]	市区的酒店	containstopword(p)⇒!(person(p) v location(p) v organization(p))
“百帮”服务中心[ORG]	“百帮”服务中心	containpunctuation(p)⇒!(person(p) v location(p) v organization(p))

only belong to one class.

We assume that the relational database contains only binary relations. Each extracted NE candidate is represented by one or more strings appearing as arguments of ground atoms in the database. The goal of NE prediction is to determine whether the candidates are entities and the types of entities (query predicates), given the evidence predicates and other relations that can be deterministically derived from the database. As we will see, despite their simplicity and consistency, these first-order formulas incorporate the essential features for NE prediction.

4 Experiments

4.1 Dataset

We used People's Daily corpus (January-Jun, 1998) in our experiments, which contains approximately 357K sentences, 156K PERs, 219K LOCs and 87K ORGs, respectively. We did some modifications on the original data to make it cleaner. We enriched some tags so that the abbreviation proper nouns are well labeled. We preprocessed some nested names to make them in better form. We also processed some person names. We enriched tags for different kinds of person names (e.g., Chinese and transliterated names) and separated consecutive person names.

4.2 The Baseline NER System

We use CRFs to build a character-based Chinese NER system, with features described in Section 2.1. To avoid overfitting, we penalized the log-likelihood by the commonly used zero-mean Gaussian prior over the parameters. In addition, we exploit clue word features which can capture non-local dependencies. This gives us a competitive baseline CRF model using both local and non-local information for Chinese NER.

For clue word features, we employ 412 career titles (e.g., 总统/President, 教授/Professor, 警察/Police), 59 family titles (e.g., 爸爸/Father, 妹妹/Sister), 33 personal pronouns (e.g., 你们/Your, 我们/We) and 109 direction words (e.g., 以北/North, 南部/South) to represent non-local information. Career titles, family titles and personal pronouns may imply a nearby PER and direction words may indicate a LOC or an ORG. Figure 3 illustrates an example of non-local dependency.

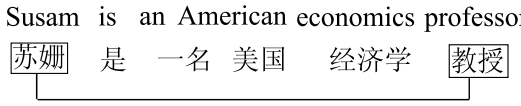


Figure 3: An Example of Non-local Dependency. The Career Title “教授” Indicates a PER “苏姗”

We do not take the advantage of using the golden-standard word segmentation and POS tagging provided in the original corpus, since such information is hardly available in real text. Instead, we use an off-the-shelf Chinese lexical analysis system, the open source ICTCLAS (Zhang *et al.*, 2003), to segment and POS tag the corpus. This module employs a hierarchical Hidden Markov Model (HHMM) and provides word segmentation, POS tagging (labels Chinese words using a set of 39 tags) and unknown word recognition. It performs reasonably well, with segmentation precision recently evaluated at 97.58%. The recall of unknown words using role tagging is over 90%.

We use one-month corpus for training and 9-day corpus for testing. Table 4 shows the experimental results.

4.3 NER System Based on Graphical Models with Logic

To test the effectiveness of our proposed model, we extract all the NEs (19,879 PERs, 25,661 LOCs and 11,590 ORGs) from the training corpus. An MLN training database, which consists of 14 predicates, 16,620 constants and 97,992 ground atoms was built.

The MLNs were trained using a Gaussian prior with zero mean and unit variance on each weight to penalize the pseudo-likelihood, and with the weights initialized at the mode of the prior (zero). During MLN learning, each formula is converted to Conjunctive Normal Form (CNF), and a weight is learned for each of its clauses. The weight

Table 4: Chinese NER by CRF Model

	Precision	Recall	$F_{\beta=1}$
Character features			
PER	92.88%	79.42%	85.62
LOC	90.95%	82.88%	86.73
ORG	88.16%	83.86%	85.96
Overall	90.92%	82.07%	86.27
Character+Word			
PER	93.27%	82.99%	87.83
LOC	91.49%	85.16%	88.21
ORG	88.94%	84.79%	86.82
Overall	91.48%	84.46%	87.83
Character+Word+POS			
PER	92.17%	90.64%	91.40
LOC	90.56%	89.74%	90.15
ORG	89.15%	85.19%	87.12
Overall	90.76%	89.13%	89.94
All features			
PER	92.12%	90.57%	91.34
LOC	90.62%	89.74%	90.18
ORG	89.72%	85.44%	87.53
Overall	90.89%	89.16%	90.02

Table 5: Chinese NER by Graphical Models with Logic

	Precision	Recall	$F_{\beta=1}$	RER
CRF Baseline				
PER	92.12%	90.57%	91.34	
LOC	90.62%	89.74%	90.18	
ORG	89.72%	85.44%	87.53	
Overall	90.89%	89.16%	90.02	
Graphical Models (GS Inference)				
PER	93.52%	93.32%	93.42	
LOC	93.19%	91.91%	92.55	
ORG	90.16%	90.71%	90.43	
Overall	92.70%	92.09%	**92.39**	23.75%
Graphical Models (ST Inference)				
PER	93.52%	93.32%	93.42	
LOC	93.19%	91.91%	92.55	
ORG	90.16%	90.71%	90.43	
Overall	92.70%	92.09%	**92.39**	23.75%
Graphical Models (MC-SAT Inference)				
PER	93.52%	93.32%	93.42	
LOC	93.19%	91.91%	92.55	
ORG	90.16%	90.71%	90.43	
Overall	92.70%	92.09%	**92.39**	23.75%
Graphical Models (MAP/MPE Inference)				
PER	92.87%	93.15%	93.01	
LOC	93.15%	91.61%	92.37	
ORG	90.56%	89.10%	89.82	
Overall	92.57%	91.58%	**92.07**	20.54%

of a clause is used as the mean of a Gaussian prior for the learned weight. These weights reflect how often the clauses are actually observed in the training data.

We extract 529 entity candidates to construct the MLN testing database, which contains 2,543 entries and these entries are used as evidence for inference. Inference is performed by grounding the minimal subset of the network required for answering the query predicates. We employed 3 MCMC algorithms: Gibbs sampling (GS), Simulated Tempering (ST) as well as MC-SAT, and the MAP/MPE algorithm for inference and the comparative NER results are shown. The probabilistic graphical models greatly outperform the CRF model stand-alone by a large margin. It can be seen from Table 5, the probabilistic graphical models integrating first-order logic improve the precision and recall for all kinds of entities, thus boosting the overall F-measure. We achieve a 23.75% relative error reduction (RER) on F-measure by using 3 MCMC algorithms and a 20.54% RER by using MAP/MPE algorithm, over an already competitive CRF baseline. We obtained the same results using GS, ST and MC-SAT algorithms. MCMC algorithms yields slightly better results than the MAP/MPE algorithm.

4.4 Significance Test

Ideally, comparisons among NER systems would control for feature sets, data preparation, training and test procedures, parameter tuning, and estimate the statistical significance of performance differences. Unfortunately, reported results sometimes leave out details needed for accurate comparisons.

We give statistical significance estimates using McNemar's paired tests [4] (Gillick and Cox, 1989) on labeling disagreements for CRF model and graphical probabilistic models that we evaluated directly.

Table 6 summarizes the correctness of the labeling decisions between the models with a 95% confidence interval (CI). These tests suggest that the graphical probabilistic models are significantly more accurate and confirm that the gains we obtained are statistically highly significant.

Table 6: McNemar's Tests on Labeling Disagreements

Null Hypothesis	95% CI	p-value
Proposed Model (GS) vs. CRFs	5.71-9.52	$< 1 \cdot 10^{-6}$
Proposed Model (ST) vs. CRFs	5.71-9.52	$< 1 \cdot 10^{-6}$
Proposed Model (MC-SAT) vs. CRFs	5.71-9.52	$< 1 \cdot 10^{-6}$
Proposed Model (MAP/MPE) vs. CRFs	4.50-7.37	$< 1 \cdot 10^{-6}$

5 Related Work

As a well-established task, Chinese NER has been studied extensively and a number of techniques for this task have been reported in the literature. Most recently, the trend in Chinese NER is to use improved machine learning approaches, or to integrate various kinds of useful evidences, features, or resources.

Fu and Luke (2005) presented a lexicalized HMM-based approach to unifying unknown word identification

[4] Most researchers refer to statistically significant as $p < 0.05$ and statistically highly significant as $p < 0.001$.

and NER as a single tagging task on a sequence of known words. Although lexicalized HMMs was shown to be superior to standard HMMs, this approach has some disadvantages: it is a purely statistical model and it suffers from the problem of data sparseness. And the model fails to tag some complicated NEs (e.g., nested ORGs) correctly due to lack of domain adaptive techniques. The F-measures of LOCs and ORGs are only 87.13 and 83.60, which show that there is still a room for improving.

A method of incorporating heuristic human knowledge into a statistical model was proposed in (Wu *et al.*, 2005). Here Chinese NER was regarded as a probabilistic tagging problem and the heuristic human knowledge was used to reduce the searching space. However, this method assumes that POS tags are golden-standard in the training data and heuristic human knowledge is often ad hoc. These drawbacks make the method unstable and highly sensitive to POS errors; and when golden-standard POS tags are not available (this is often the case), it may degrade the performance.

Cohen and Sarawagi (2004) proposed a semi-Markov model which combines a Markovian, HMM-like extraction process and a dictionary component. This process is based on sequentially classifying segments of several adjacent words. However, this technique requires that entire segments have the same class label, while our technique does not. Moreover, compared to a large-scale dictionary, our domain knowledge is much easier to obtain.

However, all the above models treat NER as classification or sequence labeling problem. To the best of our knowledge, MLNs have not been previously used for NER problem. To our knowledge, we first view Chinese NER as a *statistical relational learning* problem and exploit domain knowledge which can be concisely formulated in MLNs, allowing the training and inference algorithms to be directly applied to them.

6 Conclusion and Future Work

The contribution of this paper is three-fold. First, we formulate Chinese NER as a *statistical relational learning* problem and propose a new framework incorporating probabilistic graphical models and first-order logic for Chinese NER which achieves state-of-the-art performance. Second, We incorporate domain knowledge to capture the essential features of the NER task via MLNs, a unified framework for SRL which produces a set of weighted first-order clauses to predict new NE candidates. To the best of our knowledge, this is the first attempt at using MLNs for the NER problem in the NLP community. Third, our proposed framework can be extendable to language-independent NER, due to the simplicity of the domain knowledge we could access. Directions for future work include learning the structure of MLNs automatically and using MLNs for information extraction (e.g., entity relation extraction).

References

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, February 1999.

Andrew Borthwick. *A Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University, September 1999.

Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

Aitao Chen, Fuchun Peng, Roy Shan, and Gordon Sun. Chinese named entity recognition with conditional probabilistic models. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.

Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. Chinese named entity recognition with conditional random fields. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.

Hai Leong Chieu and Hwee Tou Ng. Named entity recognition with a maximum entropy approach. In *Proceedings of CoNLL-03*, 2003.

William W. Cohen and Sunita Sarawagi. Exploiting dictionaries in named entity extraction: Combining semi-Markov extraction processes and data integration methods. In *Proceedings of ACM-SIGKDD 2004*, 2004.

Guohong Fu and Kang-Kwong Luke. Chinese named entity recognition using lexicalized HMMs. *ACM SIGKDD Explorations Newsletter*, 7:19–25, June 2005.

Michael R. Genesereth and Nils J. Nislsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Mateo, CA, 1987.

W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, UK, 1996.

L. Gillick and Stephen Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of ICASSP-89*, pages 532–535, 1989.

Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of COLING-02*, pages 1–7, Taipei, Taiwan, 2002.

John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Enzo Marinari and Giorgio Parisi. Simulated Tempering: A new Monte Carlo scheme. *Europhysics Letters*, 19:451–458, 1992.

Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL-03*, 2003.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1988.

Fuchun Peng and Andrew McCallum. Accurate information extraction from research papers using conditional random fields. In *Proceedings of HLT-NAACL 2004*, pages 329–336, 2004.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of COLING-04*, pages 562–568, 2004.

David Pinto, Andrew McCallum, Xing Wei, and W. Bruce Croft. Table extraction using conditional random fields. In *Proceedings of ACM SIGIR-03*, 2003.

Hoifung Poon and Pedro Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proceedings of AAAI-06*, Boston, Massachusetts, July 2006. The AAAI Press.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, Geneva, Switzerland, 2004.

Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL 2003*, pages 213–220, 2003.

Maosong Sun, Changning Huang, Haiyan Gao, and Jie Fang. Identifying Chinese names in unrestricted texts. *Journal of Chinese Information Processing*, 1995.

Youzheng Wu, Jun Zhao, Bo Xu, and Hao Yu. Chinese named entity recognition based on multiple features. In *Proceedings of HLT-EMNLP 2005*, 2005.

Xiaofeng Yu, Marine Carpuat, and Dekai Wu. Boosting for Chinese named entity recognition. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.

Hua Ping Zhang, Qun Liu, Xue-Qi Cheng, Hao Zhang, and Hong Kui Yu. Chinese lexical analysis using Hierarchical Hidden Markov Model. In *2nd SIGHAN Workshop on Chinese Language Processing*, volume 17, pages 63–70, 2003.

Guodong Zhou and Jian Su. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of ACL-02*, pages 473–480, Philadelphia, USA, 2002.

Junsheng Zhou, Liang He, Xinyu Dai, and Jiajun Chen. Chinese named entity recognition with a multi-phase model. In *5th SIGHAN Workshop on Chinese Language Processing*, Australia, July 2006.

A Hybrid Feature Set based Maximum Entropy Hindi Named Entity Recognition

Sujan Kumar Saha
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
sujan.kr.saha@gmail.com

Sudeshna Sarkar
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
shudeshna@gmail.com

Pabitra Mitra
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
pabitra@gmail.com

Abstract

We describe our effort in developing a Named Entity Recognition (NER) system for Hindi using Maximum Entropy (MaxEnt) approach. We developed a NER annotated corpora for the purpose. We have tried to identify the most relevant features for Hindi NER task to enable us to develop an efficient NER from the limited corpora developed. Apart from the orthographic and collocation features, we have experimented on the efficiency of using gazetteer lists as features. We also worked on semi-automatic induction of context patterns and experimented with using these as features of the MaxEnt method. We have evaluated the performance of the system against a blind test set having 4 classes - Person, Organization, Location and Date. Our system achieved a f-value of 81.52%.

1 Introduction

Named Entity Recognition involves locating and classifying the names in text. NER is an important task, having applications in Information Extraction (IE), question answering, machine translation and in most other NLP applications.

NER systems have been developed for English and few other languages with high accuracies. These systems take advantage of large amount of Named Entity (NE) annotated corpora and other NER resources. However when we started working on a NER system for Hindi, we did not have any NER annotated corpora for Hindi, neither did we have access to any comprehensive gazetteer list.

In this work we have identified suitable features for the Hindi NER task. Orthography features, the suffix and prefix information, as well as information about the sorrounding words and their tags are used to develop a Maximum Entropy (MaxEnt) based Hindi NER system. Additionally, we have acquired gazetteer lists for Hindi and used these gazetteers in the Maximum Entropy (MaxEnt) based Hindi NER system. We also worked on semi-automatically learning of context pattern for identifying names. These context pattern rules have been integrated into the MaxEnt based NER system, leading to a high accuracy.

The paper is organized as follows. A brief survey of different techniques used for the NER task in different languages and domains are presented in Section 2. The MaxEnt based NER system is described in Section 3. Various features used in NER are then discussed. Next we present the experimental results and related discussions. Finally Section 8 concludes the paper.

2 Previous Work

A variety of techniques has been used for NER. The two major approaches to NER are:

1. Linguistic approaches.
2. Machine Learning based approaches.

The linguistic approaches typically use rules manually written by linguists. There are several rule-based NER systems, containing mainly lexicalized

grammar, gazetteer lists, and list of trigger words, which are capable of providing 88%-92% f-measure accuracy for English (Grishman, 1995; McDonald, 1996; Wakao et al., 1996).

The main disadvantages of these rule-based techniques are that these require huge experience and grammatical knowledge of the particular language or domain and these systems are not transferable to other languages or domains.

Machine Learning (ML) based techniques for NER make use of a large amount of NE annotated training data to acquire high level language knowledge. Several ML techniques have been successfully used for the NER task of which Hidden Markov Model (Bikel et al., 1997), Maximum Entropy (Borthwick, 1999), Conditional Random Field (Li and Mccallum, 2004) are most common. Combinations of different ML approaches are also used. Srihari et al. (2000) combines Maximum Entropy, Hidden Markov Model and handcrafted rules to build an NER system.

NER systems use gazetteer lists for identifying names. Both the linguistic approach (Grishman, 1995; Wakao et al., 1996) and the ML based approach (Borthwick, 1999; Srihari et al., 2000) use gazetteer lists.

The linguistic approach uses hand-crafted rules which needs skilled linguistics. Some recent approaches try to learn context patterns through ML which reduce amount of manual labour. Talukder et al.(2006) combined grammatical and statistical techniques to create high precision patterns specific for NE extraction. An approach to lexical pattern learning for Indian languages is described by Ekbal and Bandopadhyay (2007). They used seed data and annotated corpus to find the patterns for NER.

The NER task for Hindi has been explored by Cucerzan and Yarowsky in their language independent NER work which used morphological and contextual evidences (Cucerzan and Yarowsky, 1999). They ran their experiment with 5 languages - Romanian, English, Greek, Turkish and Hindi. Among these the accuracy for Hindi was the worst. For Hindi the system achieved 41.70% f-value with a very low recall of 27.84% and about 85% precision. A more successful Hindi NER system was developed by Wei Li and Andrew Mccallum (2004) using Conditional Random Fields (CRFs) with feature induction. They were able to achieve 71.50% f-value using a training set of size 340k words. In Hindi the maximum accuracy is achieved by (Kumar and Bhattacharyya, 2006). Their Maximum Entropy Markov Model (MEMM) based model gives 79.7% f-value.

3 Maximum Entropy Based Model

We have used a Maximum Entropy model to build the NER in Hindi. MaxEnt is a flexible statistical model which assigns an outcome for each token based on its history and features. MaxEnt computes the probability $p(o|h)$ for any o from the space of all possible outcomes O, and for every h from the space of all possible histories H. A history is all the conditioning data that enables one to assign probabilities to the space of outcomes. In NER, history can be viewed as all information derivable from the training corpus relative to the current token. The computation of $p(o|h)$ in MaxEnt depends on a set of features, which are helpful in making predictions about the outcome. The features may be binary-valued or multi-valued. For instance, one of our features is: the current token is a part of the surname list; how likely is it to be part of a person name. Formally, we can represent this feature as follows:

$$f(h,o) = \begin{cases} 1 & \text{if } w_i \text{ in surname list and } o = \text{person} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Given a set of features and a training corpus, the MaxEnt estimation process produces a model in which every feature f_i has a weight α_i. We can compute the conditional probability as (Pietra et al., 1997):

$$p(o|h) = \frac{1}{Z(h)} \prod_i {\alpha_i}^{f_i(h,o)} \quad (2)$$

$$Z(h) = \sum_o \prod_i {\alpha_i}^{f_i(h,o)} \quad (3)$$

So the conditional probability of the outcome is the product of the weights of all active features, normalized over the products of all the features. For our development we have used a Java based opennlp MaxEnt toolkit[1] to get the probability values of

[1]www.maxent.sourceforge.net.

a word belonging to each class. That is, given a sequence of words, the probability of each class is obtained for each word. To find the most probable tag corresponding to each word of a sequence, we can choose the tag having the highest class conditional probability value. But this method is not good as it might result in an inadmissible output tag.

Some tag sequences should never happen. To eliminate these inadmissible sequences we have made some restrictions. Then we used a beam search algorithm with a beam of length 3 with these restrictions.

The training data for this task is composed of about $243K$ words which is collected from the popular daily Hindi newspaper "Dainik Jagaran". This corpus has been manually annotated and has about 16,482 NEs. In this development we have considered 4 types of NEs, these are $Person$(P), $Location$(L), $Organization$(O) and $Date$(D). To recognize entity boundaries each name class N is subdivided into 4 sub-classes, i.e., N_Begin, $N\_Continue$, N_End, and N_Unique. Hence, there are a total of 17 classes including 1 class for not-name. The corpus contains $6,298$ Person, $4,696$ Location, $3,652$ Organization and $1,845$ Date entities.

4 Features for Hindi NER

Machine learning approaches like MaxEnt, CRF etc. make use of different features for identifying the NEs. Orthographic features (like capitalization, decimal, digits), affixes, left and right context (like previous and next words), NE specific trigger words, gazetteer features, POS and morphological features etc. are generally used for NER. In English and some other languages, capitalization features play an important role as NEs are generally capitalized for these languages. Unfortunately this feature is not applicable for Hindi. Also Indian person names are more diverse, lots of common words having other meanings are also used as person names. These make difficult to develop a NER system on Hindi. Li and Mccallum (2004) used the entire word text, character n-grams (n = 2, 3, 4), word prefix and suffix of lengths 2, 3 and 4, and 24 Hindi gazetteer lists as atomic features in their Hindi NER. Kumar and Bhattacharyya (2006) used word features (suffixes, digits, special characters), context features, dictionary features, NE list features etc. in their MEMM based Hindi NER system. In the following we have discussed about the features we have identified and used to develop the Hindi NER system.

4.1 Feature Description

The features which we have identified for Hindi Named Entity Recognition are:

Static Word Feature: The previous and next words of a particular word are used as features. The previous m words $(w_{i-m}...w_{i-1})$ to next n words $(w_{i+1}...w_{i+n})$ can be treated. During our experiment different combinations of previous 4 to next 4 words are used.

Context Lists: Context words are defined as the frequent words present in a word window for a particular class. We compiled a list of the most frequent words that occur within a window of $w_{i-3}...w_{i+3}$ of every NE class. For example, location context list contains the words like '$jAkara$[2]' (going to), '$desha$' (country), '$rAjadhAnI$' (capital) etc. and person context list contains '$kahA$' (say), '$prdhAnama.ntrI$' (prime minister) etc. For a given word, the value of this feature corresponding to a given NE type is set to 1 if the window $w_{i-3}...w_{i+3}$ around the w_i contains at last one word from this list.

Dynamic NE tag: Named Entity tags of the previous words $(t_{i-m}...t_{i-1})$ are used as features.

First Word: If the token is the first word of a sentence, then this feature is set to 1. Otherwise, it is set to 0.

Contains Digit: If a token 'w' contains digit(s) then the feature $ContainsDigit$ is set to 1. This feature is helpful for identifying company product names (e.g. 06WD1992), house number (e.g. C226) etc.

Numerical Word: For a token 'w' if the word is a numerical word i.e. a word denoting a number (e.g. eka (one), do (two), $tina$ (three) etc.) then the feature $NumWord$ is set to 1.

Word Suffix: Word suffix information is helpful to identify the named NEs. Two types of suffix features have been used. Firstly a fixed length word suffix of the current and surrounding words are used

[2]All Hindi words are written in italics using the 'Itrans' transliteration.

as features. Secondly we compiled lists of common suffixes of person and place names in Hindi. For example, '$pura$', '$bAda$', '$nagara$' etc. are location suffixes. We used two binary features corresponding to the lists - whether a given word has a suffix from the list.

Word Prefix: Prefix information of a word may be also helpful in identifying whether it is a NE. A fixed length word prefix of current and surrounding words are treated as a features.

Parts-of-Speech (POS) Information: The POS of the current word and the surrounding words may be useful feature for NER. We have access to a Hindi POS pagger developed at IIT Kharagpur which has an accuracy about 90%. The tagset of the tagger contains 28 tags. We have used the POS values of the current and surrounding tokens as features.

We realized that the detailed POS tagging is not very relevant. Since NEs are noun phrases, the noun tag is very relevant. Further the postposition following a name may give a clue to the NE type. So we decided to use a coarse-grained tagset with only three tags - nominal (Nom), postposition (PSP) and other (O).

The POS information is also used by defining several binary features. An example is the $NomPSP$ binary feature. The value of this feature is defined to be 1 if the current token is nominal and the next token is a PSP.

5 Enhancement using Gazetteer Feature

Lists of names of various types are helpful in name identification. We have compiled some specialized name lists from different web sources. But the names in these lists are in English, not in Hindi. So we have transliterated these English name lists to make them useful for our Hindi NER task.

For the transliteration we have build a 2-phase transliteration module. We have defined an intermediate alphabet containing 34 characters. English names are transliterated to this intermediate form using a map-table. Hindi strings are also transliterated to the intermediate alphabet form using a different map-table. For a English-Hindi string pair, if transliterations of the both strings are same, then we conclude that one string is the transliteration of the other. This transliteration module works with 91.59% accuracy.

Using the transliteration approach we have constructed 8 lists. Which are, month name and days of the week (*40*)[3], organization end words list (*92*), person prefix words list (*123*), list of common locations (*80*), location names list (*17,600*), first names list (*9722*), middle names list (*35*), surnames list (*1800*).

The lists can be used in name identification in various ways. One way is to check whether a token is in any list. But this approach is not good as it has some limitations. Some words may present in two or more gazetteer lists. For example, '$bangAlora$' is in surnames list and also in location names list. Confusions arise to make decisions for these words. Some words are in gazetteer lists but sometimes these are used in text as not-name entity. For example, '$gayA$' is in location list but sometimes the word is used as verb in text and makes confusion. These limitations might be reduced if the contexts are considered.

We have used these gazetteer lists as features of MaxEnt. We have prepared several binary features which are defined as whether a given word is in a particular list. For example, a binary feature $FirstName$ is 1 for a particular token 't' if 't' is in the first name list.

6 Context Pattern based Features

Context patterns are helpful for identifying NEs. As manual identification of context patterns takes much manual labour and linguistic knowledge, we have developed a module for semi-automatically learning of context pattern. The summary of the context pattern learning module is given follows:

1. Collect some seed entities (E) for each class.
2. For each seed entity e in E, from the corpus find context string(C) comprised of n tokens before e, a placeholder for the class instance and n tokens after e. [We have used $n = 3$] This set of tokens form initial pattern.
3. Search the pattern in the corpus and find the coverage and precision.
4. Discard the patterns having low precision.

[3]The italics integers in brackets indicate the size of the lists.

5. Generalize the patterns by dropping one or more tokens to increase coverage.

6. Find best patterns having good precision and coverage.

The quality of a pattern is measured by precision and coverage. Precision is the ratio of correct identification and the total identification, when the particular pattern is used to identify of NEs of a specific type from a raw text. Coverage is the amount of total identification. We have given more importance to precision and we have marked a pattern as $effective$ if the precision is more than 95%. The method is applied on an un-annotated text having 4887011 words collected from "Dainik Jagaran" and context patterns are learned. These context patterns are used as features of MaxEnt in the Hindi NER system. Some example patterns are:

1. mukhyama.ntrI <PER> Aja

2. <PER> ne kahA ki

3. rAjadhAnI <LOC> me

7 Evaluation

We have evaluated the system using a blind test corpus of 25K words, which is distinct from the training corpus. The accuracies are measured in terms of the f-measure, which is the weighted harmonic mean of precision and recall. Here we can mention that we have evaluated the performance of the system on actual NEs. That means the system annotates the test data using 17 tags, similar to the training data. During evaluation we have merged the sub-tags of a particular entity to get a complete NEs and calculated the accuracies. At the end of section 7.1 we have also mentioned the accuracies if evaluated on the tags. A number of experiments are conducted considering various combinations of features to identify the best feature set for the Hindi NER task.

7.1 Baseline

The baseline performance of the system without using gazetteer and context patterns are presented in Table 1. They are summarized below.

While experimenting with static word features, we have observed that a window of previous two words to next two words ($W_{i-2}...W_{i+2}$) gives best results. But when several other features are combined then single word window ($W_{i-1}...W_{i+1}$) performs better. Similarly we have experimented with suffixes of different lengths and observed that the suffixes of length $\leq$ 2 gives the best result for the Hindi NER task. In using POS information, we have observed that the coarse-grained POS tagger information is more effective than the finer-grained POS values. A feature set, combining finer-grained POS values, surrounding words and previous NE tag, gives a f-value of 70.39%. But when the coarse-grained POS values are used instead of the

Feature	**Class**	**F-value**
f1 = Word, NE Tag	PER	63.33
	LOC	69.56
	ORG	58.58
	DAT	91.76
	TOTAL	69.64
f2 = Word, NE Tag, Suffix (≤ 2)	PER	69.75
	LOC	75.8
	ORG	59.31
	DAT	89.09
	TOTAL	73.42
f3 = Word, NE Tag, Suffix (≤ 2), Prefix	PER	70.61
	LOC	71
	ORG	59.31
	DAT	89.09
	TOTAL	72.5
f4 = Word, NE Tag, Digit, Suffix (≤ 2)	PER	70.61
	LOC	75.8
	ORG	60.54
	DAT	93.8
	TOTAL	74.26
f5 = Word, NE Tag, POS	PER	64.25
	LOC	71
	ORG	60.54
	DAT	89.09
	TOTAL	70.39
f6 = Word, NE Tag, Suffix (≤ 2), Digit, $NomPSP$	PER	72.26
	LOC	78.6
	ORG	51.36
	DAT	92.82
	TOTAL	**75.6**

Table 1: F-values for different features

finer-grained POS values, the f-value is increased to 74.16%. The most interesting fact we have observed that more complex features do not guarantee to achieve better results. For example, a feature set combined with current and surrounding words, previous NE tag and fixed length suffix information, gives a f-value 73.42%. But when prefix information are added the f-value decreased to 72.5%. The highest accuracy achieved by the system is 75.6% f-value without using gazetteer information and context patterns.

The results in Table 1 are obtained by evaluating on the actual NEs. But when the system is evaluated on the tags the f-value increases. For f6, the accuracy achieved on actual NEs is 75.6%, but if evaluated on tags, the value increased to 77.36%. Similarly, for f2, the accuracy increased to 75.91% if evaluated on tags. The reason is the NEs containing 3 or more words, are subdivided to N-begin, N-continue (1 or more) and N-end. So if there is an error in any of the subtags, the total NE becomes an error. We observed many cases where NEs are partially identified by the system, but these are considered as $error$ during evaluation.

7.2 Using Gazetteer Lists and Context Patterns

Next we add gazetteer and context patterns as features in our MaxEnt based NER system. In Table 2 we have compared the results after addition of gazetteer information and context patterns with previous results. While experimenting we have observed that gazetteer lists and context patterns are capable of increasing the performance of our baseline system. That is tested on all the baseline feature sets. In Table 2 the comparison is shown for only two features - f2 and f6 which are defined in Table 1. It may be observed that the relative advantage of using both gazetteer and context patterns together over using them individually is not much. For example, when gazetteer information are added with f2, the f-value is increased by 6.38%, when context patterns are added the f-value is increased by 6.64%., but when both are added the increment is 7.27%. This may be due to the fact that both gazetteer and context patterns lead to the same identifications. Using the comprehensive feature set (using gazetteer information and context patterns) the MaxEnt based NER system achieves the maximum f-value of 81.52%.

Feature	**F-value**				
	Class	**No Gaz or Pat**	**With Gaz**	**With Pat**	**With Gaz and Pat**
f2	PER	69.75	74.2	75.61	76.03
	LOC	75.8	82.02	79.94	82.02
	ORG	59.31	72.61	73.4	74.63
	DAT	89.09	94.29	95.32	95.32
	TOTAL	73.42	79.8	80.06	80.69
f6	PER	72.26	76.03	75.61	78.41
	LOC	78.6	82.02	80.49	83.26
	ORG	51.36	72.61	74.1	75.43
	DAT	92.82	94.28	95.87	96.5
	TOTAL	75.6	80.24	80.37	**81.52**

Table 2: F-values for different features with gazetteers and context patterns

8 Conclusion

We have shown that our MaxEnt based NER system is able to achieve a f-value of 81.52%, using a hybrid set of features including traditional NER features augmented with gazetteer lists and extracted context patterns. The system outperforms the existing NER systems in Hindi.

Feature selection and feature clustering might lead to further improvement of performance and is under investigation.

9 Acknowledgement

The work is partially funded by Microsoft Research India.

References

Bikel Daniel M., Miller Scott, Schwartz Richard and Weischedel Ralph. 1997. Nymble: A High Performance Learning Name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 194–201.

Borthwick Andrew. 1999. A Maximum Entropy Approach to Named Entity Recognition. *Ph.D. thesis, Computer Science Department, New York University*.

Cucerzan Silviu and Yarowsky David. 1999. Language Independent Named Entity Recognition Combining

Morphological and Contextual Evidence. In *Proceedings of the Joint SIGDAT Conference on EMNLP and VLC 1999*, pages 90–99.

Ekbal A. and Bandyopadhyay S. 2007. Lexical Pattern Learning from Corpus Data for Named Entity Recognition. In *Proceedings of International Conference on Natural Language Processing (ICON), 2007*.

Grishman Ralph. 1995. The New York University System MUC-6 or Where's the syntax? In *Proceedings of the Sixth Message Understanding Conference*.

Kumar N. and Bhattacharyya Pushpak. 2006. Named Entity Recognition in Hindi using MEMM. In *Technical Report, IIT Bombay, India.*.

Li Wei and McCallum Andrew. 2004. Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction (Short Paper). *ACM Transactions on Computational Logic*.

McDonald D. 1996. Internal and external evidence in the identification and semantic categorization of proper names. In *B. Boguraev and J. Pustejovsky, editors, Corpus Processing for Lexical Acquisition*, pages 21–39.

Pietra Stephen Della, Pietra Vincent Della and Lafferty John. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

Srihari R., Niu C. and Li W. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. In *Proceedings of the sixth conference on Applied natural language processing*.

Talukdar Pratim P., Brants T., Liberman M., and Pereira F. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*.

Wakao T., Gaizauskas R. and Wilks Y. 1996. Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings of COLING-96*.

An effective method of using Web based information for Relation Extraction

Yong Wai Keong, Stanley
Institute for Inforcomm Research
21 Heng Mui Keng Terrace,
Singapore 119613
wkyong@i2r.a-star.edu.sg

Su Jian
Institute for Inforcomm Research
21 Heng Mui Keng Terrace,
Singapore 119613
sujian@i2r.a-star.edu.sg

Abstract

We propose a method that incorporates paraphrase information from the Web to boost the performance of a supervised relation extraction system. Contextual information is extracted from the Web using a semi-supervised process, and summarized by skip-bigram overlap measures over the entire extract. This allows the capture of local contextual information as well as more distant associations. We observe a statistically significant boost in relation extraction performance.

We investigate two extensions, thematic clustering and hypernym expansion. In tandem with thematic clustering to reduce noise in our paraphrase extraction, we attempt to increase the coverage of our search for paraphrases using hypernym expansion.

Evaluation of our method on the ACE 2004 corpus shows that it out-performs the baseline SVM-based supervised learning algorithm across almost all major ACE relation types, by a margin of up to 31%.

1 Introduction and motivation

In this paper, we shall be primarily dealing with the sort of relations defined in the NIST's Automatic Content Extraction program, specifically for the Relation Detection and Characterization (RDC) task (Doddington et al., 2004). These are links between two entities mentioned in the same sentence, and further restrict our consideration to those relationships clearly supported by evidence in the scope of the same document.

The ACE's annotators mark all mentions of relations where there is a direct syntactic connection between the entities, i.e. when one entity mention modifies another one, or when two entity mentions are arguments of the same event. Relations between entities that are implied in the text but which do not satisfy either requirement are considered to be implicit, and are marked only once.

Our work sits squarely in the realm of work on regular IE done by (Zelenko et al., 2003; Zhou et al., 2005; Chen et al., 2006). Here, the corpus of interest is a well defined set of texts, such as news articles, and we have to detect and classify all appearances of relations from a set of given relation types in the documents. In line with assumptions in the related work, we assert that the differences in the markup for implicit and explicit relations does not significantly affect our performance.

Supervised learning methods have proved to be some of the most effective for regular IE. They do however, need large volumes of expensively annotated examples to perform robustly. As a result, even the large ACE compilation has deficiencies in the number of instances available for some of the relation types. (Zhang et al., 2005) reports an F-score of 50% for categories of intermediate size and only 30% for the least common relation types. It appears that there are hard limits on the performance of relation extraction systems as long as they have to rely solely on information in the training set.

We were thus inspired to explore how one could

exploit the Web, the largest raw text collection freely available, for regular IE. In this paper, we detail the ways one can fruitfully employ relation specific sentences retrieved from the Web with a semi-supervised labeling approach. Most importantly we show how the output from such an approach can be combined with existing knowledge gleaned from supervised learning to improve the performance of relation extraction significantly.

2 Related Work and Differences

To our knowledge, there is no previous work that exploits the information from a large raw text corpus like the Web to improve supervised relation extraction. In the spirit of the work done by (Shinyama and Sekine, 2003; Bunescu and Mooney, 2007), we are trying to collect clusters of paraphrases for given relation mentions. Briefly, since the same relation can be expressed in many ways, the information we may learn about that relation in any single sentence is very limited. The idea then is to alleviate this bias by collecting many paraphrases of the same relation instance into clusters when we train our system.

Shinyama generalizes the expressions using part-of-speech information and dependency tree similarity into generic templates. Bunescu's work uses a relation kernel on subsquences of words developed in (Bunescu and Mooney, 2005). We observed that both approaches suffer from low recall despite the attempts to generalize the subsequences and templates probably because they rely on local context only.

Based on our observation, we looked for a way to use our clusters without losing non-local information about the sentences. Bag-of-words or unigram representations of our paraphrase clusters are easy to compute, but information about word ordering is lost. Hence, we settled on the use of a skip-bigram representation of our relation clusters instead.

Skip-bigrams (Lin and Och, 2004) are pairs of words in sentence order allowing for gaps in between. Longer gaps capture far-flung associations between words, while short gaps of between 1 and 4 capture local structure. In using them, we do not restrict ourselves to context centered around the entity mentions. Another advantage of using skip-bigrams is that we can capture some extra-sentential information since we are no longer restricted by the ability to generate dependency structures within a single sentence as Shinyama is.

Using skip-bigrams, we can assess the similarity of a particular new relation mention instance against the relation clusters we collect in training. We can then compute a likelihood that we combine with the predictions of the supervised learning algorithm for final classification.

Two possible extensions to the basic method stated above were examined.

A central problem with the paraphrase collection approach when applied to an open corpus is noise. As pointed out by (Bunescu and Mooney, 2007), even though the same entities co-occur in multiple sentences, they are not necessarily linked by the same relationship in all of them. The problem is exacerbated when the open corpus we look at contains documents from heterogenous domains. Indeed, we cannot even assume that the predominant relation that holds between two entities in the set of sentences is the relation of interest to us.

One means of combating this is suggested by (Bunescu and Mooney, 2007). They re-weight the importance of word features in their model to reduce topic drift. We try a different solution based on the thematic clustering of sentences. Sentences extracted from the raw corpus are mapped to a vector space and partitioned into different clusters using the Partitioning Around Medoids algorithm (Kaufmann and Rousseeuw, 1987). Sentences in the clusters closest to the original relation mention instance are more likely to embody the same relationship. Hence we retain such clusters, while discarding the rest. As the relation we wish to recover may not be the predominant one, the cluster that is retained is also often not the largest one.

Another problem identified by Shinyama is that the same entity may itself be referred to in different ways. If the form used in the original relation mention is uncommon, then few paraphrases will be found. For instance, "'President Bush"' may be referred to as "'Dubya"' by a writer. Searching for sentences online with the word "'Dubya"' and the other entity participating in the relation is likely to result in a collection heavily biased towards the originator of the nickname. Shinyama's solution is to use a limited form of co-reference resolution to re-

place these forms with a more general noun phrase. As co-reference resolution is itself an unreliable process, we suggest the use of hypernym substitution instead.

In subsequent sections, we will outline the structure of our system, examine the experimental evidence of its viability using the ACE program data, and finish with a discussion of the extensions.

3 Overall structure

Our system is organized very naturally into two main phases, a learning or training phase, followed by a usage or testing phase. The learning phase is subdivided into two parallel paths, reflecting the hybrid nature of our algorithm.

Fully supervised learning based on annotations takes place in tandem with a semi-supervised algorithm that captures the paraphrase clusters using entity mention instances. We will combine the models from both the supervised learning and the semi-supervised algorithm using a meta-classifier trained a different subset of the data.

3.1 Learning Procedure

Our goal is to acquire as much contextual information from the available annotations as possible via our supervised learner and expand on that using Web based information found by our semi-supervised algorithm.

We constructed our fully supervised learner according to the specifications for the system developed by (Zhou et al., 2005). It utilizes a feature based framework with a Support Vector Machine (SVM) classifier (Cortes and Vapnik, 1995). We support the same set of features as Zhou, namely: local word position features, entity type, base phrase chunking features, dependency and parse tree features as well as semantic information like country names and a list of trigger words. In our current work, we use Michael Collins' (Collins, 2003) parser for syntactic information.

Sentence boundary detection, chunking, and parsing are done as preprocessing steps before we begin our learning.

Given a sentence with the relation mention instance, the semi-supervised method goes through the following five stages:

1. From the list of entitites marked in the sentence, generate all possible pairings as candidates. We pick one of these candidates and proceed to the next step.

2. Gather hypernyms for each entity mention using Wordnet synsets and generate all possible combinations of entity pairs from the two sets.

3. Find sentences from the Web that mention both entities.

4. Cluster the sentences found using k-medoids, filter out noise and retain only the cluster that the original relation mention would be assigned to.

5. Collate all clusters by relation type and generate a skip-bigram index for each relation type.

We will spend the rest of this section on the details of each stage.

3.1.1 Extracting entity mentions and gathering hypernyms

The process starts when we receive a relation mention and the sentence it was found in, for instance in the following sentence from the ACE 2004 corpus: *"As president of Sotheby's, she often conducted the biggest, the highest profile auctions."*

From the annotation, we find that the two entities of interest are *president* and *Sotheby's* in the *EMP-ORG Executive* relation, which we could represent as the predicate $EMP-ORG\ Executive(em_1, em_2)$. Since our aim is to find as many instances of semantically similar sentences as possible, we want to lessen the negative impact of quirky spelling or naming in a given sentence. Hence we do a hypernym search in Wordnet to find more general terms for our entities and create list of similarly related entities $(em_1\prime, em_2\prime)$ etc. If the mention is a named entity, we do without the hypernym expansion and use coreference information (when available) to find the most common substring amongst the mentions for the same entity.

In this example, it might result in the four pairs show in Table 1.

Table 1: Examples of entity pairs after hypernym expansion

President	Sotheby's
Chief Executive	Sotheby's
Decision maker	Sotheby's
leader	Sotheby's

Table 2: Examples of extracted text from Google

The 40 year old former president travels incognitio to Sotheby's
Brooks was named president of Sotheby's Inc.
Subsidiary President at Sotheby's...
Bill Ruprecht, chief executive of Sotheby's, agreed that September 11 had been...
The Duke will also remain leader of Sotheby's Germany...

3.1.2 Web search

(Geleijnse and Korst, 2006) use Google as their search engine for extracting surface patterns from Web documents. We use the same procedure here to find our paraphrases. For each pair of arguments, we create a boolean query string $em_1 * em_2$, and submit it to Google. The query will find documents where mentions of em_1 are separated from em_2 by any number of words. We restrict the language to English.

Google returns a two line extract from the documents that match our boolean query. The extracts are generally those lines where the key query terms are most densely collocated. These are parsed and obviously nonsensical sentences are discarded based on the occurrence of words from a list of stop words. If a group of sentences are very similar, we choose a single representative and discard the rest. For every remaining sentence, we normalize them by removing extraneous HTML tags. Some examples of extracts found are listed in Table 2.

3.1.3 Cluster, filter and collate

In general, the collection of sentence extracts we have at the end of the previous stage are likely to be about a diverse range of topics. As we are only interested in the subset that is most compatible with the thematic content of our original relation mention, we will have to filter away unrelated extracts. For example, the *EMP-ORG Executive* relation does not hold in the sentence: *"The 40 year old former president travels incognito to Sotheby's"*.

We make use of the K-medoids method by (Kaufmann and Rousseeuw, 1987). The terms from all sentences are extracted and their frequency in each sentence is computed. Each sentence is now mapped as a vector of frequencies in the space of the terms that we observed. The resulting vectors are stored as a large matrix. Picking a random partition of the sentences as our starting point, we assign some sentences to be the cluster centers, and iteratively refine the clusters based on a distance minimization heuristic.

Through some preliminary experiments, we find that K-medoids based clustering with 5 classes produced the most consistent results. From a list of excerpts, our algorithm culls the sentences that belonged to the 4 irrelevant clusters and produces the excerpts which capture the original relationship best.Since the quality of the partitions produced by the algorithm is sensitive to the initial random start, we do this process twice with different configurations and take the union of the two clusters as our final result.

The best excerpts are stored with accompanying meta-data about the originating training relation instance in what we call *pseudo-documents*. We group the pseudo-documents in our database by the relation label that the instance pairs were given. Thus we end up with several bags of pseudo-documents, where each bag corresponds to a single relation type of interest.

For computational efficiency, we generate an inverted hash-index for the pseudo-documents. Our skip-bigrams act as the keys and the records are lists of meta-data nodes. Each node records the sentence that the bigram is observed in, the relation type of that sentence, and the position of the bigram.

All we need now is a means of measuring the similarity of a new relation mention instance with the bags of pseudo-documents to assign a relation label to it.

3.1.4 Skip-bigrams

As discussed in the introduction, instead of generalizing our bags of documents into patterns or relation extraction kernels, we create skip-bigram indices. There are several advantages in doing so.

Skip-bigrams are easy to compute relative to the dependency trees or subsequence kernels used in (Shinyama and Sekine, 2003) or (Bunescu and Mooney, 2005). Moreover, we can tune the number of gaps allowed to capture long-distance word dependency information, which is not done by the other approaches because it is relatively more expensive for them to do so, due to the combinatorial explosion. In addition, as compared to Bunescu's subsequence approach which needs 4 words to match, bigrams are far less likely to be sparse in the document space.

Since we relied upon skip-bigrams in our queries to Google, it is only natural that we use it again in assessing the similarity of two pseudo-documents. Each pseudo-document is really an extractive summary of online articles about the same topic, with the same entities. The degree of overlap between two pseudo-documents is a good measure of their thematic overlap.

Now that we have a metric, that still leaves the question of our matching heuristic open. Do we automatically assign a test instance the relation label of the sentence with the highest skip-bigram overlap? This naive approach is problematic. In general, longer sentences will have more bigrams and hence higher probability of overlapping with other sentences. We could normalize the bigram overlap score by the length of the sentences, but here we leave the optimization to a machine learner.

Another possible heuristic is to pick the relation label whose bag has the highest number of matching bigrams with our test instance. Again, this will be biased, but now towards bags with larger numbers of pseudo-documents. A last possibility is to look at the total number of sentences that have bigram matches, and weight the overlap score higher for those with more sentence matches.

Therefore, instead of designing the heuristic explicitly, we use a validation set to observe the statistical correlations of each of the three possible heuristics we discussed above. We train an additional model, using an SVM to choose the weights for each heuristic automatically.

Accordingly, we do the following for each validation instance, V, and its pseudo-document P.

For each extracted sentence j in pseudo-document P, we look up the database of pseudo-documents from our training set, and compute the skip-bigram similarity with every single sentence. We have a skip-bigram similarity score for every single sentence in the database with respect to V. The scores are collated according to the relation classes.

For each relation class we generate three numbers, TopS, Matching docs, and Total. Using the notation by (Lin and Och, 2004), we denote the skip-bigram overlap between two sentences X and Y as $Skip2(X, Y)$. For the i^{th} relation, C_i is the set of all pseudo-documents in our training set of that relation type.

$$TopS = \max_{Y \in C_i, j \in P} Skip2(P_j, Y) \quad (1)$$

$$Matching = \sum_{Y \in C_i} \sum_{j \in P} I[Skip2(P_j, Y) > 0] \quad (2)$$

$$Total = \sum_{Y \in C_i} \sum_{j \in P} Skip2(P_j, Y) \quad (3)$$

The three figures provide a summary of the best sentence level match, and the overall relation level overlap in terms of the number of sentences and number of overlaps. As an illustration, we consider the case where we have two sentences in our pseudo-document $P = P_1, P_2$ and a relation RX. We compute $Skip2(P_1, Y)$ and $Skip2(P_2, Y)$ by looking up the skip-bigrams in the database for RX and aggregating over sentences. Let's assume that $|RX| = 3$ and only $Skip2(P_1, Y_1) = 2$, $Skip2(P_1, Y_3) = 5$, $Skip2(P_2, Y_3) = 4$ are non-zero. Then TopS for instance V is 5. Matching will be 2 since only Y_1 and Y_3 have overlaps with elements of P. The Total is simply $2 + 5 + 4 = 11$.

3.2 Combining supervised with semi-supervised models

After the preceding steps, we have a trained SVM model based, and our skip-bigram index from the semi-supervised procedure. In this section, we will describe a method of combining these into a better classifier.

The validation data we left aside earlier is sent through our system with the relation labels removed. Each entity pair in this validation set has a corresponding pseudo-document and a file with numerical features for the SVM model.

An instance V is scored by Zhou's SVM classifier, which assigns a relation tag, V_{ST} to it. In parallel, the skip-bigram assessment results in $\{TopS, Matching, Total\}$ scores for each of the relation classes. We treat the tag and numbers as features for training another SVM, which we shall refer to as SVM_C. This is our final meta-classifier for relation extraction on the tenth of the original data set aside for testing. The meta-classifier may also be used for completely new data.

4 Experimentation

We use the ACE corpus provided by the LDC from 2004 to train and evaluate our system. There are 674 annotated text documents and 9683 relation instances in the set.

Starting with a single training set provided by the user, we split that into three parts: the majority (80%) is used for the learning phase, one tenth is used for the validation during construction of the combined model, and the remaining tenth is used for testing. We ran a series of experiments, using five-fold cross-validation.

Unlike typical cross-validation, the fifth that we set aside is further sub-divided into two parts as we stated before. Half is used when we construct the hybrid model merging supervised and semi-supervised paths, and the remainder is used for the actual testing and evaluation.

We used the 6 main relation types defined in the annotation for the ACE dataset: ART, EMP-ORG, GPE-AFF, OTHER-AFF, PER-SOC, and PHYS. We computed three measures of the effectiveness, the recall (R), precision (P) and F-1 score (F-1).

4.1 Comparison against the baseline

The first set of experiments we shall discuss compares the overall system against our baseline. The baseline system is implemented as a feature extraction module on top of a set of binary SVM classifiers. A primary classifier is used to separate the candidates without any relation of interest from the rest. Secondary classifiers for each relation type are then used to partition the positive candidates by voting. The performance of our baseline classifier when tested on the ACE2003 dataset is statistically indistinguishable from that reported by Zhou et al. in (Zhou et al., 2005).

Drilling down to the level of individual relation classes as shown below, we note that the meta-classifier performs better than the baseline on all but one of the relations. This might be due to the inherent ambiguity of the OTHER-AFF class.

Relation	Ratio	System	R	P	F-1
ART	5.6	Hybrid	0.48	0.73	0.59
		baseline	0.29	0.43	0.34
EMP-ORG	40.0	Hybrid	0.78	0.75	0.76
		baseline	0.67	0.83	0.73
GPE-AFF	11.7	Hybrid	0.49	0.59	0.53
		baseline	0.36	0.56	0.45
OTHER-AFF	3.4	Hybrid	0.14	0.18	0.16
		baseline	0.18	0.59	0.28
PER-SOC	9.7	Hybrid	0.63	0.80	0.70
		baseline	0.32	0.5	0.39
PHYS	29.4	Hybrid	0.75	0.59	0.66
		baseline	0.40	0.64	0.49

The hybrid system has slightly lower precision on the two largest relation classes, EMP-ORG and PHYS, but higher recall, resulting in better F-scores on both types. Finally, note that on the three intermediate sized classes, ART, PER-SOC, and GPE-AFF, the recall and precision were both higher. The results suggest that the Web information does improve recall substantially, but affects precision in cases where there already is a substantial amount of training data. It confirms our original assertion that the hybrid approach works well for mid-sized relation classes, where the amount of training data is not enough for the supervised system to perform optimally.

We use the T-test to see if our improvements over the baseline were significant at a 0.05 level. To summarize, recall for the PHYS, PER-SOC, GPE-AFF and EMP-ORG relations was improved significantly. The difference in precision is significant for the PER-SOC, and OTHER-AFF classes, while F-1 score differences for the PER-SOC and PHYS relation classes at 0.00085, 0.032 respectively were both significant.

4.2 Testing hypernym expansion and clustering

Subsequent experiments were aimed at quantifying the contribution of hypernym expansion and thematic clustering to our hybrid system. We ran a 2 factorial experiment with four of our five folds, where we take the hypernym expansion and clustering as treatments. Since we have more than one feature being tested, and we wish to observe the relative contribution of each factor, we used an ANOVA test instead of T-tests (Montgomery, 2005).

Our intuitive justification for hypernym expansion was that recall would be boosted for relation types where the name entities tend to be overly specific in the training corpus. Place names, personal names and the object names are obvious targets. Indeed, we noted that the recall did increase in absolute terms on average for the ART, PER-SOC and PHYS relation types (about 3% for each), but declined slightly (about 0.5%) for the rest. Overall however, the size of the effects was too small to be statistically significant. This suggests that other methods of term generalization may be needed to achieve a larger effect.

Next, we looked at the contribution of clustering. Our initial experiments showed that k-medoids with 5 clusters was able to produce very precise clusters. However, it would be at the expense of some of the potential gain in recall form Web extracts.

Our experiments shows that clustering does indeed lower the potential recall. However, the hoped for improvement in precision was observed only in the PER-SOC (6%) and GPE-AFF (0.7%) relations. This suggests that the effect of name entities having multiple relations is concentrated in the classes of named entities related to Persons and GPEs. Again, the size of the effects was not statistically significant.

A more thorough investigation of clustering techniques with different settings for k and different algorithms will be needed before we can make stronger statements.

5 Discussion and Conclusion

We have presented a hybrid approach to relation extraction that incorporates Web based information successfully to boost the performance of state-of-the-art supervised feature based systems. Evaluation on the ACE corpus shows that our skip-bigram based relevance measures for finding the right paraphrase in our Web extract database are very effective.

While our analysis shows that the addition of clustering and hypernym expansion to the skip-bigram based process is not statistically significant, we have indications that the effect on recall and precision is positive for certain relation classes.

In future work, we will examine improvements to the clustering algorithm to reduce the impact on recall. We will look at alternative ways of attacking the problem of name entity generalization and assess the impact of methods like co-reference resolution in the same ANOVA framework.

Acknowledgment

This research is supported by a Specific Targeted Research Project (STREP) of the European Union's 6th Framework Programme within IST call 4, Bootstrapping Of Ontologies and Terminologies STtragegic REsearch Project (BOOTStrep).

References

Razvan C. Bunescu and Raymond J. Mooney. 2005. Subsequence kernels for relation extraction. In *NIPS*.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583, Prague, Czech Republic, June. Association for Computational Linguistics.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zheng-Yu Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In *ACL*. The Association for Computer Linguistics.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. volume 20, pages 273–297.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program. tasks, data and evaluation.

G. Geleijnse and J. Korst. 2006. Learning effective surface text patterns for information extraction. In *Proceedings of the EACL 2006 workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 1 – 8, Trento, Italy,

April. The Association for Computational Linguistics. http://acl.ldc.upenn.edu/eacl2006/ws06_atem.pdf.

L. Kaufmann and P. J. Rousseeuw. 1987. Clustering by means of medoids. In Y. Dodge, editor, *Statistical Data Analysis based on the L1 Norm*, pages 405–416, Amsterdam. Elsevier/North Holland.

Chin Y. Lin and Franz J. Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of Coling 2004*, pages 501–507, Geneva, Switzerland, Aug FebruaryMarch–Aug FebruaryJuly. COLING.

Douglas C. Montgomery. 2005. *Design and analysis of experiments*. John Wiley And Sons.

Yusuke Shinyama and Satoshi Sekine. 2003. Paraphrase acquisition for information extraction. In Kentaro Inui and Ulf Hermjakob, editors, *Proceedings of the Second International Workshop on Paraphrasing*, pages 65–71.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *IJCNLP*, pages 378–389.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL*. The Association for Computer Linguistics.

Minimally Supervised Learning of Semantic Knowledge from Query Logs

Mamoru Komachi
Nara Institute of Science and Technology
8916-5 Takayama
Ikoma, Nara 630-0192, Japan
mamoru-k@is.naist.jp

Hisami Suzuki
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
hisamis@microsoft.com

Abstract

We propose a method for learning semantic categories of words with minimal supervision from web search query logs. Our method is based on the *Espresso* algorithm (Pantel and Pennacchiotti, 2006) for extracting binary lexical relations, but makes important modifications to handle query log data for the task of acquiring semantic categories. We present experimental results comparing our method with two state-of-the-art minimally supervised lexical knowledge extraction systems using Japanese query log data, and show that our method achieves higher precision than the previously proposed methods. We also show that the proposed method offers an additional advantage for knowledge acquisition in an Asian language for which word segmentation is an issue, as the method utilizes no prior knowledge of word segmentation, and is able to harvest new terms with correct word segmentation.

1 Introduction

Extraction of lexical knowledge from a large collection of text data with minimal supervision has become an active area of research in recent years. Automatic extraction of relations by exploiting recurring patterns in text was pioneered by Hearst (1992), who describes a bootstrapping procedure for extracting words in the hyponym (*is-a*) relation, starting with three manually given lexico-syntactic patterns. This idea of learning with a minimally supervised bootstrapping method using surface text patterns was subsequently adopted for many tasks, including relation extraction (e.g., Brin, 1998; Riloff and Jones, 1999; Pantel and Pennacchiotti, 2006) and named entity recognition (e.g., Collins and Singer, 1999; Etzioni et al., 2005).

In this paper, we describe a method of learning semantic categories of words using a large collection of Japanese search query logs. Our method is based on the *Espresso* algorithm (Pantel and Pennacchiotti, 2006) for extracting binary lexical relations, adapting it to work well on learning unary relations from query logs. The use of query data as a source of knowledge extraction offers some unique advantages over using regular text.

- Web search queries capture the interest of search users directly, while the distribution of the Web documents do not necessarily reflect the distribution of what people search (Silverstein et al., 1998). The word categories acquired from query logs are thus expected to be more useful for the tasks related to search.
- Though user-generated queries are often very short, the words that appear in queries are generally highly relevant for the purpose of word classification.
- Many search queries consist of *keywords*, which means that the queries include word segmentation specified by users. This is a great source of knowledge for learning word boundaries for those languages whose regularly written text does not indicate word boundaries, such as Chinese and Japanese.

Although our work naturally fits into the larger goal of building knowledge bases automatically from text, to our knowledge we are the first to explore the use of Japanese query logs for the purpose of minimally supervised semantic category acquisition. Our work is similar to Sekine and Suzuki (2007), whose goal is to augment a manually created dictionary of named entities by finding

	# of seed	Target	# of iteration	Corpus	Language
Sekine & Suzuki	~600	Categorized NEs	1	Query log	English
Basilisk	10	Semantic lexicon	∞	MUC-4	English
Espresso	~10	Semantic relations	∞	TREC	English
Tchai	5	Categorized words	∞	Query log	Japanese

Table 1: Summary of algorithms

contextual patterns from English query logs. Our work is different in that it does not require a full-scale list of categorized named entities but a small number of seed words, and iterates over the data to extract more patterns and instances. Recent work by Paşca (2007) and Paşca and Van Durme (2007) also uses English query logs to extract lexical knowledge, but their focus is on learning attributes for named entities, a different focus from ours.

2 Related Work

In this section, we describe three state-of-the-art algorithms of relation extraction, which serve as the baseline for our work. They are briefly summarized in Table 1. The goal of these algorithms is to learn target *instances*, which are the words belonging to certain categories (e.g., *cat* for the Animal class), or in the case of relation extraction, the pairs of words standing in a particular relationship (e.g., *pasta*::*food* for *is-a* relationship), given the *context patterns* for the categories or relation types found in source data.

2.1 Pattern Induction

The first step toward the acquisition of instances is to extract context patterns. In previous work, these are surface text patterns, e.g., *X such as Y*, for extracting words in an *is-a* relation, with some heuristics for finding the pattern boundaries in text. As we use query logs as the source of knowledge, we simply used everything but the instance string in a query as the pattern for the instance, in a manner similar to Paşca et al. (2006). For example, the seed word *JAL* in the query "*JAL+flight_schedule*" yields the pattern "*#+flight_schedule*".[1] Note that we perform no word segmentation or boundary detection heuristics in identifying these patterns, which makes our approach fast and robust, as the segmentation errors introduce noise in extracted patterns, especially when the source data contains many out of vocabulary items.

The extracted context patterns must then be assigned a score reflecting their usefulness in extracting the instances of a desired type. Frequency is a poor metric here, because frequent patterns may be extremely *generic*, appearing across multiple categories. Previously proposed methods differ in how to assign the desirability scores to the patterns they find and in using the score to extract instances, as well as in the treatment of generic patterns, whose precision is low but whose recall is high.

2.2 Sekine and Suzuki (2007)'s Algorithm

For the purpose of choosing the set of context patterns that best characterizes the categories, Sekine and Suzuki (2007) report that none of the conventional co-occurrence metrics such as tf.idf, mutual information and chi-squared tests achieved good results on their task, and propose a new measure, which is based on the number of different instances of the category a context *c* co-occurs with, lized by its token frequency for all categories:

$$Score(c) = f_{type} \log \frac{g(c)}{C}$$

$$g(c) = f_{type}(c) / F_{inst}(c)$$

$$C = f_{type}(ctop1000) / F_{inst}(ctop1000)$$

where f_{type} is the type frequency of instance terms that *c* co-occurs with in the category, F_{inst} is the token frequency of context *c* in the entire data and *ctop1000* is the 1000 most frequent contexts. Since they start with a large and reliable named entity dictionary, and can therefore use several hundred seed terms, they simply used the top-*k* highest-scoring contexts and extracted new named entities once and for all, without iteration. Generic patterns receive low scores, and are therefore ignored by this algorithm.

2.3 The *Basilisk* Algorithm

Thelen and Riloff (2002) present a framework called *Basilisk*, which extracts semantic lexicons

[1] # indicates where the instance occurs in the query string, and + indicates a white space in the original Japanese query. The underscore symbol (_) means there was originally no white space; it is used merely to make the translation in English more readable.

[2] The manual classification assigns only one category

for multiple categories. It starts with a small set of seed words and finds all patterns that match these seed words in the corpus. The bootstrapping process begins by selecting a subset of the patterns by the *R*log*F* metric (Riloff, 1996):

$$R\log F(pattern_i) = \frac{F_i}{N_i} \cdot \log(F_i)$$

where F_i is the number of category members extracted by $pattern_i$ and N_i is the total number of instances extracted by $pattern_i$. It then identifies instances by these patterns and scores each instance by the following formula:

$$AvgLog(word_i) = \frac{\sum_{j=1}^{P_i} \log(F_j + 1)}{P_i}$$

where P_i is the number of patterns that extract $word_i$. They use the average logarithm to select instances to balance the recall and precision of generic patterns. They add five best instances to the lexicon according to this formula, and the bootstrapping process starts again. Instances are cumulatively collected across iterations, while patterns are discarded at the end of each iteration.

2.4 The *Espresso* Algorithm

We will discuss the *Espresso* framework (Pantel and Pennacchiotti, 2006) in some detail because our method is based on it. It is a general-purpose, minimally supervised bootstrapping algorithm that takes as input a few seed instances and iteratively learns surface patterns to extract more instances. The key to *Espresso* lies in its use of generic patterns: Pantel and Pennacchiotti (2006) assume that correct instances captured by a generic pattern will also be instantiated by some *reliable patterns*, which denote high precision and low recall patterns.

Espresso starts from a small set of seed instances of a binary relation, finds a set of surface patterns *P*, selects the top-*k* patterns, extracts the highest scoring *m* instances, and repeats the process. *Espresso* ranks all patterns in *P* according to reliability r_π, and retains the top-*k* patterns for instance extraction. The value of *k* is incremented by one after each iteration.

The reliability of a pattern *p* is based on the intuition that a reliable pattern co-occurs with many reliable instances. They use pointwise mutual information (PMI) and define the reliability of a pattern *p* as its average strength of association across each input instance *i* in the set of instances *I*, weighted by the reliability of each instance *i*:

$$r_\pi(p) = \frac{\sum_{i \in I} \left(\frac{pmi(i,p)}{\max_{pmi}} \cdot r_i(i) \right)}{|I|}$$

where $r_i(i)$ is the reliability of the instance *i* and $\max_{pmi}$ is the maximum PMI between all patterns and all instances. The PMI between instance $i = \{x,y\}$ and pattern *p* is estimated by:

$$pmi(i,p) = \log \frac{|x,p,y|}{|x,*,y||*,p,*|}$$

where $|x,p,y|$ is the frequency of pattern *p* instantiated with terms *x* and *y* (recall that *Espresso* is targeted at extracting binary relations) and where the asterisk represents a wildcard. They multiplied $pmi(i,p)$ with the discounting factor suggested in Pantel and Ravichandran (2004) to alleviate a bias towards infrequent events.

The reliability of an instance is defined similarly: a reliable instance is one that associates with as many reliable patterns as possible.

$$r_i(i) = \frac{\sum_{p \in P} \left(\frac{pmi(i,p)}{\max_{pmi}} \cdot r_\pi(p) \right)}{|P|}$$

where $r_\pi(p)$ is the reliability of pattern *p*, and *P* is the set of surface patterns. Note that $r_i(i)$ and $r_\pi(p)$ are recursively defined: the computation of the pattern and instance reliability alternates between performing pattern reranking and instance extraction. Similarly to *Basilisk*, instances are cumulatively learned, but patterns are discarded at the end of each iteration.

3 The *Tchai* Algorithm

In this section, we describe the modifications we made to *Espresso* to derive our algorithm called *Tchai*.

3.1 Filtering Ambiguous Instances and Patterns

As mentioned above, the treatment of high-recall, low-precision generic patterns (e.g., *#+map*, *#+animation*) present a challenge to minimally supervised learning algorithms due to their amguity. In the case of semantic category acquisition, the problem of ambiguity is exacerbated, because not only the acquired patterns, but also the instances can be highly ambiguous. For example,

once we learn an ambiguous instance such as *Pokemon*, it will start collecting patterns for multiple categories (e.g., Game, Animation and Movie), which is not desirable.

In order to control the negative effect of the generic patterns, *Espresso* introduces a confidence metric, which is similar but separate from the reliability measure, and uses it to filter out the generic patterns falling below a confidence threshold. In our experiments, however, this metric did not produce a score that was substantially different from the reliability score. Therefore, we did not use a confidence metric, and instead opted for not ing ambiguous instances and patterns, where we define *ambiguous instance* as one that induces more than 1.5 times the number of patterns of viously accepted reliable instances, and *ambiguous* (or *generic*) *pattern* as one that extracts more than twice the number of instances of previously accepted reliable patterns. As we will see in Section 4, this modification improves the precision of the extracted instances, especially in the early stages of iteration.

3.2 Scaling Factor in Reliability Scores

Another modification to the *Espresso* algorithm to reduce the power of generic patterns is to use *local* $\max_{pmi}$ instead of *global* $\max_{pmi}$. Since PMI ranges $[-\infty, +\infty]$, the point of dividing $pmi(i,p)$ by $\max_{pmi}$ in *Espresso* is to normalize the reliability to [0, 1]. However, using PMI directly to estimate the reliability of a pattern when calculating the reliability of an instance may lead to unexpected results because the absolute value of PMI is highly variable across instances and patterns. We define the *local* $\max_{pmi}$ of the reliability of an instance to be the absolute value of the maximum PMI *for a given instance*, as opposed to taking the maximum for all instances in a given iteration. Local $\max_{pmi}$ of the reliability of a pattern is defined in the same way. As we show in the next section, this modification has a large impact on the effectiveness of our algorithm.

3.3 Performance Improvements

Tchai, unlike *Espresso*, does not perform the pattern induction step between iterations; rather, it simply recomputes the reliability of the patterns induced at the beginning. Our assumption is that fairly reliable patterns will occur with at least one of the seed instances if they occur frequently enough in query logs. Since pattern induction is computationally expensive, this modification reduces the computation time by a factor of 400.

Category	Seeds (with English translation)
Travel	jal, ana, jr, じゃらん(jalan), his
Finance	みずほ銀行(Mizuho Bank), 三井住友銀行 (SMBC), jcb, 新生銀行 (Shinsei Bank), 野村證券(Nomura Securities)

Table 2: Seed instances for Travel and Financial Services categories

4 Experiment

In this section, we present an empirical comparison of *Tchai* with the systems described in Section 2.

4.1 Experimental Setup

Query logs: The data source for instance extraction is an anonymized collection of query logs submitted to Live Search from January to February 2007, taking the top 1 million unique queries. Queries with garbage characters are removed. Almost all queries are in Japanese, and are accompanied by their frequency within the logs.

Target categories: Our task is to learn word categories that closely reflect the interest of web search users. We believe that a useful categorization of words is task-specific, therefore we did not start with any externally available ontology, but chose to start with a small number of seed words. For our task, we were given a list of 23 categories relevant for web search, with a manual classification of the 10,000 most frequent search words in the log of December 2006 (which we henceforth refer to as the *10K list*) into one of these categories.[2] For evaluation, we chose two of the categories, Travel and Financial Services: Travel is the largest category containing 712 words of the 10K list (as all the location names are classified into this category), while Financial Services was the smallest, containing 240 words.

Systems: We compared three different systems described in Section 2 that implement an iterative algorithm for lexical learning:

[2] The manual classification assigns only one category per word, which is not optimal given how ambiguous the category memberships are. However, it is also very difficult to reliably perform a multi-class categorization by hand.

	10K list		Not in 10K list
	Travel	Not Travel	
Travel	280	17	251
Not Travel	0	7	125

Table 3: Comparison with manual annotation: Travel category

	10K list		Not in 10K list
	Finance	Not Finance	
Finance	41	30	30
Not Finance	0	5	99

Table 4: Comparison with manual annotation: Financial Services category

- *Basilisk*: The algorithm by (Thelen and Riloff, 2002) described in Section 2.
- *Espresso*: The algorithm by (Pantel and Pennacchiotti, 2006) described in Sections 2 and 3.
- *Tchai*: The *Tchai* algorithm described in this paper.

For each system, we gave the same seed instances. The seed instances are the 5 most frequent words belonging to these categories in the 10K list; they are given in Table 2. For the Travel category, "jal" and "ana" are airline companies, "jr" stand for Japan Railways, "jalan" is an online travel information site, and "his" is a travel agency. In the Finance category, three of them are banks, and the other two are a securities company and a credit card firm. *Basilisk* starts by extracting 20 patterns, and adds 100 instances per iteration. *Espresso* and *Tchai* start by extracting 5 patterns and add 200 instances per iteration. *Basilisk* and *Tchai* iterated 20 times, while *Espresso* iterated only 5 times due to computation time.

4.2 Results

4.2.1 Results of the *Tchai* algorithm

Tables 3 and 4 are the results of the *Tchai* algorithm compared to the manual classification. Table 3 shows the results for the Travel category. The precision of *Tchai* is very high: out of the 297 words classified into the Travel domain that were also in the 10K list, 280 (92.1%) were learned rectly.[3] It turned out that the 17 instances that represent the precision error were due to the ambiguity of hand labeling, as in 東京ディズニーランド 'Tokyo Disneyland', which is a popular travel destination, but is classified as Entertainment in the manual annotation. We were also able to correctly learn 251 words that were not in the 10K list according to manual verification; we also harvested 125 new words "incorrectly" into the Travel domain, but these words include common nouns related to Travel, such as 釣り 'fishing' and レンタカー 'rental car'. Results for the Finance domain show a similar trend, but fewer instances are extracted.

Sample instances harvested by our algorithm are given in Table 5. It includes subclasses of travel-related terms, for some of which no seed words were given (such as Hotels and Attractions). We also note that segmentation errors are entirely absent from the collected terms, demonstrating that query logs are in fact excellently suited for acquiring new words for languages with no explicit word segmentation in text.

Type	Examples (with translation)
Place	トルコ (Turkey), ラスベガス (Las Vegas), バリ島 (Bali Island)
Travel agency	Jtb, トクー (www.tocoo.jp), yahoo (Yahoo ! Travel), net cruiser
Attraction	ディズニーランド (Disneyland), usj (Universal Studio Japan)
Hotel	帝国ホテル(Imperial Hotel), リッツ(Ritz Hotel)
Transportation	京浜急行(Keihin Express), 奈良交通(Nara Kotsu Bus Lines)

Table 5: Extracted Instances

4.2.2 Comparison with *Basilisk* and *Espresso*

Figures 1 and 2 show the precision results comparing *Tchai* with *Basilisk* and *Espresso* for the Travel and Finance categories. *Tchai* outperforms *Basilisk* and *Espresso* for both categories: its precision is constantly higher for the Travel category, and it achieves excellent precision for the Finance category, especially in early iterations. The differences in behavior between these two categories are due to the inherent size of these domains. For the

[3] As the 10K list contained 712 words in the Travel category, the recall against that list is fairly low (~40%). The primary reason for this is that all location names are classified as Travel in the 10K list, and 20 iterations are not enough to enumerate all frequent location names. Another reason is that the 10K list consists of *queries* but our algorithm extracts *instances* – this sometimes causes a mismatch, e.g.,*Tchai* extracts リッツ 'Ritz' but the 10K list contains リッツホテル 'Ritz Hotel'.

	# of inst.	Precision	Rel.recall
Basilisk	651	63.4	1.26
Espresso	500	65.6	1.00
Tchai	680	80.6	1.67

Table 6: Precision (%) and relative recall: Travel domain

	# of inst.	Precision	Rel.recall
Basilisk	278	27.3	0.70
Espresso	704	15.2	1.00
Tchai	223	35.0	0.73

Table 7: Precision (%) and relative recall: Financial Services domain

smaller Finance category, *Basilisk* and *Espresso* both suffered from the effect of generic patterns such as #ホームページ 'homepage' and #カード 'card' in early iterations, whereas *Tchai* did not select these patterns.

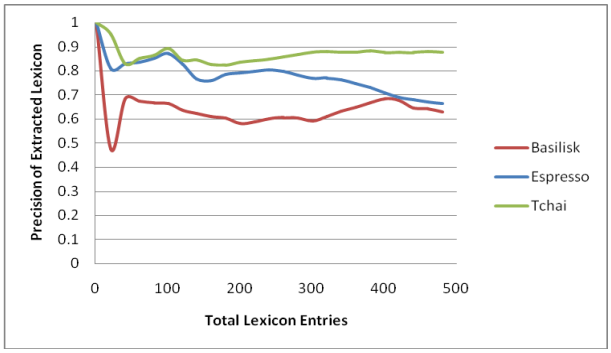


Figure 1: *Basilisk*, *Espresso* vs. *Tchai*: Travel

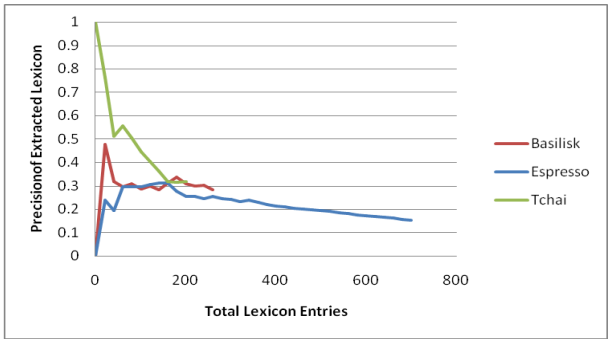


Figure 2: *Basilisk*, *Espresso* vs. *Tchai*: Finance

Comparing these algorithms in terms of recall is more difficult, as the complete set of words for each category is not known. However, we can estimate the *relative* recall given the recall of another system. Pantel and Ravichandran (2004) defined *relative recall* as:

$$R_{A|B} = \frac{R_A}{R_B} = \frac{C_A/C}{C_B/C} = \frac{C_A}{C_B} = \frac{P_A \times |A|}{P_B \times |B|}$$

where $R_{A|B}$ is the relative recall of system A given system B, C_A and C_B are the number of correct instances of each system, and C is the number of true correct instances. C_A and C_B can be calculated by using the precision, P_A and P_B, and the number of instances from each system. Using this formula, we estimated the relative recall of each system relative to *Espresso*. Tables 6 and 7 show that *Tchai* achieved the best results in both precision and relative recall in the Travel domain. In the Finance domain, *Espresso* received the highest relative call but the lowest precision. This is because *Tchai* uses a filtering method so as not to select generic patterns and instances.

Table 8 shows the context patterns acquired by different systems after 4 iterations for the Travel domain.[4] The patterns extracted by *Basilisk* are not entirely characteristic of the Travel category. For example, "*p#sonic*" and "*google+#lytics*" only match the seed word "ana", and are clearly irrelevant to the domain. *Basilisk* uses token count to estimate the score of a pattern, which may explain the extraction of these patterns. Both *Basilisk* and *Espresso* identify location names as context patterns (e.g., #東京 'Tokyo', #九州 'Kyushu'), which may be too generic to be characteristic of the domain. In contrast, *Tchai* finds context patterns that are highly characteristic, including terms related to transportation (#+格安航空券 'discount plane ticket', #マイレージ 'mileage') and accommodation (#+ホテル 'hotel').

4.2.3 Contributions of *Tchai* components

In this subsection, we examine the contribution of each modification to the *Espresso* algorithm we made in *Tchai*.

Figure 3 illustrates the effect of each modification proposed for the *Tchai* algorithm in Section 3 on the Travel category. Each line in the graph corresponds to the *Tchai* algorithm with and without the modification described in Sections 3.1 and 3.2. It shows that the modification to the $\max_{pmi}$ function (purple) contributes most significantly to the improved accuracy of our system. The filtering of generic patterns (green) does not show

[4] Note that *Basilisk* and *Espresso* use context patterns only for the sake of collecting instances, and are not interested in the patterns per se. However, they can be quite useful in characterizing the semantic categories they are acquired for, so we chose to compare them here.

System	Sample Patterns (with English translation)
Basilisk	#東日本(*east_japan*), #西日本(*west_japan*), *p#sonic*, #時刻表(*timetable*), #九州(*Kyushu*), #+マイレージ(*mileage*), #バス(*bus*), *google+#lytics*, #+料金(*fare*), #+国内(*domestic*), #ホテル(*hotel*)
Espresso	#バス(*bus*), 日本#(*Japan*), #ホテル(*hotel*), #道路(*road*), #イン(*inn*), フジ#(*Fuji*), #東京(*Tokyo*), #料金(*fare*), #九州(*Kyushu*), #時刻表(*timetable*), #+旅行(*travel*), #+名古屋(*Nagoya*)
Tchai	#+ホテル(*hotel*), #+ツアー(*tour*), #+旅行(*travel*), #予約(*reserve*), #+航空券(*flight_ticket*), #+格安航空券(*discount_flight_titcket*), #マイレージ(*mileage*), 羽田空港+#(*Haneda Airport*)

Table 8: Sample patterns acquired by three algorithms

a large effect in the precision of the acquired instances for this category, but produces steadily better results than the system without it.

Figure 4 compares the original *Espresso* algorithm and the modified *Espresso* algorithm which performs the pattern induction step only at the beginning of the bootstrapping process, as described in Section 3.3. Although there is no significant difference in precision between the two systems, this modification greatly improves the computation time and enables efficient extraction of instances. We believe that our choice of the seed instances to be the most frequent words in the category produces sufficient patterns for extracting new instances.

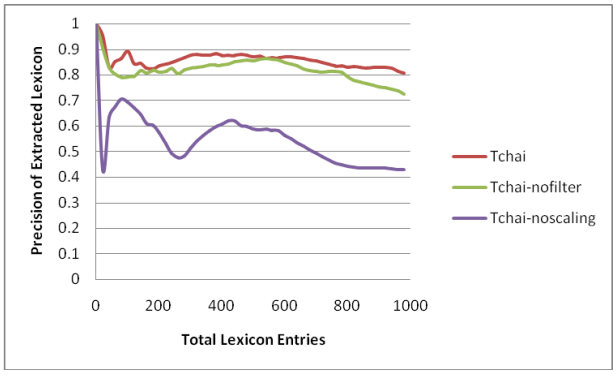


Figure 3: System precision w/o each modification

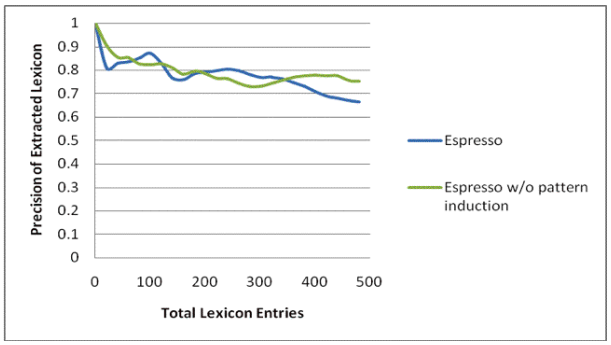


Figure 4: Modification to the pattern induction step

5 Conclusion

We proposed a minimally supervised bootstrapping algorithm called *Tchai*. The main contribution of the paper is to adapt the general-purpose *Espresso* algorithm to work well on the task of learning semantic categories of words from query logs. The proposed method not only has a superior performance in the precision of the acquired words into semantic categories, but is faster and collects more meaningful context patterns for characterizing the categories than the unmodified *Espresso* algorithm. We have also shown that the proposed method requires no pre-segmentation of the source text for the purpose of knowledge acquisition.

Acknowledgements

This research was conducted during the first author's internship at Microsoft Research. We would like to thank the colleagues at Microsoft Research, especially Dmitriy Belenko and Christian König, for their help in conducting this research.

References

Sergey Brin. 1998. Extracting Patterns and Relations from the World Wide Web. WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT '98. pp. 172-183.

Michael Collins and Yoram Singer. 1999. Unsupervised Models for Named Entity Classification. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. pp. 100-110.

Oren Etzioni, Michael Cafarella, Dong Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*. 165(1). pp. 91-134.

Marti Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the*

Fourteenth International Conference on Computational Linguistics. pp 539-545.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*. pp. 113-120.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically Labeling Semantic Classes. *Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL-04)*. pp. 321-328.

Marius Paşca. 2004. Acquisition of Categorized Named Entities for Web Search. *Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM-04)*. pp. 137-145.

Marius Paşca. 2007. Organizing and Searching the World Wide Web of Fact – Step Two: Harnessing the Wisdom of the Crowds. *Proceedings of the 16th International World Wide Web Conference (WWW-07)*. pp. 101-110.

Marius Paşca and Benjamin Van Durme. 2007. What You Seek is What You Get: Extraction of Class Attributes from Query Logs. *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*. pp. 2832-2837.

Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits and Alpa Jain. 2006. Organizing and Searching the World Wide Web of Facts – Step One: the One-Million Fact Extraction Challenge. *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*. pp. 1400-1405.

Ellen Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. pp. 1044-1049.

Ellen Riloff and Rosie Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. *Proceedings of the Sixteenth National Conference on Artificial Intellligence (AAAI-99)*. pp. 474-479.

Satoshi Sekine and Hisami Suzuki. 2007. Acquiring Ontological Knowledge from Query Logs. *Proceedings of the 16th international conference on World Wide Web*. pp. 1223-1224.

Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. 1998. Analysis of a Very Large AltaVista Query Log. *Digital SRC Technical Note* #1998-014.

Michael Thelen and Ellen Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. *Proceedings of Conference on Empirical Methods in Natural Language Processing*. pp. 214-221.

Learning a Stopping Criterion for Active Learning for Word Sense Disambiguation and Text Classification

Jingbo Zhu Huizhen Wang
Natural Language Processing Lab
Northeastern University
Shenyang, Liaoning, P.R.China, 110004
Zhujingbo@mail.neu.edu.cn
wanghuizhen@mail.neu.edu.cn

Eduard Hovy
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
hovy@isi.edu

Abstract

In this paper, we address the problem of knowing when to stop the process of active learning. We propose a new statistical learning approach, called minimum expected error strategy, to defining a stopping criterion through estimation of the classifier's expected error on future unlabeled examples in the active learning process. In experiments on active learning for word sense disambiguation and text classification tasks, experimental results show that the new proposed stopping criterion can reduce approximately 50% human labeling costs in word sense disambiguation with degradation of 0.5% average accuracy, and approximately 90% costs in text classification with degradation of 2% average accuracy.

1 Introduction

Supervised learning models set their parameters using given labeled training data, and generally outperform unsupervised learning methods when trained on equal amount of training data. However, creating a large labeled training corpus is very expensive and time-consuming in some real-world cases such as word sense disambiguation (WSD).

Active learning is a promising way to minimize the amount of human labeling effort by building an system that automatically selects the most informative unlabeled example for human annotation at each annotation cycle. In recent years active learning has attracted a lot of research interest, and has been studied in many natural language processing (NLP) tasks, such as text classification (TC) (Lewis and Gale, 1994; McCallum and Nigam, 1998), chunking (Ngai and Yarowsky, 2000), named entity recognition (NER) (Shen *et al.*, 2004; Tomanek *et al.*, 2007), part-of-speech tagging (Engelson and Dagan, 1999), information extraction (Thompson et al., 1999), statistical parsing (Steedman et al., 2003), and word sense disambiguation (Zhu and Hovy, 2007).

Previous studies reported that active learning can help in reducing human labeling effort. With selective sampling techniques such as *uncertainty sampling* (Lewis and Gale, 1994) and *committee-based sampling* (McCallum and Nigam, 1998), the size of the training data can be significantly reduced for text classification (Lewis and Gale, 1994; McCallum and Nigam, 1998), word sense disambiguation (Chen, et al. 2006; Zhu and Hovy, 2007), and named entity recognition (Shen *et al.*, 2004; Tomanek *et al.*, 2007) tasks.

Interestingly, deciding when to stop active learning is an issue seldom mentioned issue in these studies. However, it is an important practical topic, since it obviously makes no sense to continue the active learning procedure until the whole corpus has been labeled. How to define an adequate stopping criterion remains an unsolved problem in active learning. In principle, this is a problem of estimation of classifier effectiveness (Lewis and Gale, 1994). However, in real-world applications, it is difficult to know when the classifier reaches its maximum effectiveness before all unlabeled examples have been annotated. And when the unlabeled data set becomes very large, full annotation is almost impossible for human annotator.

In this paper, we address the issue of a stopping criterion for active learning, and propose a new statistical learning approach, called *minimum ex-*

pected error strategy, that defines a stopping criterion through estimation of the classifier's expected error on future unlabeled examples. The intuition is that the classifier reaches maximum effectiveness when it results in the lowest expected error on remaining unlabeled examples. This proposed method is easy to implement, involves small additional computation costs, and can be applied to several different learners, such as Naive Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVMs) models. Comparing with the confidence-based stopping criteria proposed by Zhu and Hovy (2007), experimental results show that the new proposed stopping criterion achieves better performance in active learning for both the WSD and TC tasks.

2 Active Learning Process and Problem of General Stopping Criterion

2.1 Active Learning Process

Active learning is a two-step semi-supervised learning process in which a small number of labeled samples and a large number of unlabeled examples are first collected in the initialization stage, and a close-loop stage of query and retraining is adopted. The purpose of active learning is to minimize the amount of human labeling effort by having the system in each cycle automatically select for human annotation the most informative unannotated case.

Procedure: Active Learning Process
Input: initial small training set L, and pool of unlabeled data set U
Use L to train the initial classifier C (i.e. a classifier for uncertainty sampling or a set of classifiers for committee-based sampling)
Repeat

- Use the current classifier C to label all unlabeled examples in U
- Based on active learning rules R such as uncertainty sampling or committee-based sampling, present m top-ranked unlabeled examples to oracle H for labeling
- Augment L with the m new examples, and remove them from U
- Use L to retrain the current classifier C

Until the predefined stopping criterion SC is met.

Figure 1. Active learning process

In this work, we are interested in selective sampling for pool-based active learning, and focus on *uncertainty sampling* (Lewis and Gale, 1994). The key point is how to measure the uncertainty of an unlabeled example, in order to select a new example with maximum uncertainty to augment the training data. The maximum uncertainty implies that the current classifier has the least confidence in its classification of this unlabeled example x. The well-known *entropy* is a good uncertainty measurement widely used in active learning:

$$UM(x) = -\sum_{y \in Y} P(y \mid x) \log P(y \mid x) \qquad (1)$$

where $P(y|x)$ is the *a posteriori* probability. We denote the output class $y \in Y = \{y_1, y_2, \ldots, y_k\}$. UM is the uncertainty measurement function based on the entropy estimation of the classifier's posterior distribution.

2.2 General Stopping Criteria

As shown in Fig. 1, the active learning process repeatedly provides the most informative unlabeled examples to an oracle for annotation, and update the training set, until the predefined stopping criterion SC is met. In practice, it is not clear how much annotation is sufficient for inducing a classifier with maximum effectiveness (Lewis and Gale, 1994). This procedure can be implemented by defining an appropriate stopping criterion for active learning.

In active learning process, a general stopping criterion SC can be defined as:

$$SC_{AL} = \begin{cases} 1 & effectiveness(C) \geq \theta \\ 0 & otherwise, \end{cases} \qquad (2)$$

where θ is a user predefined constant and the function *effectiveness(C)* evaluates the effectiveness of the current classifier. The learning process ends only if the stopping criterion function SC_{AL} is equal to 1. The value of constant θ represents a tradeoff between the cost of annotation and the effectiveness of the resulting classifier. A larger θ would cause more unlabeled examples to be selected for human annotation, and the resulting classifier would be more robust. A smaller θ means the resulting classifier would be less robust, and less unlabeled examples would be selected to annotate.

In previous work (Shen *et al.*, 2004; Chen *et al.*, 2006; Li and Sethi, 2006; Tomanek *et al.*, 2007), there are several common ways to define the func-

tion *effectiveness(C)*. First, previous work always used a simple stopping condition, namely, when the training set reaches desirable size. However, it is almost impossible to predefine an appropriate size of desirable training data guaranteed to induce the most effective classifier. Secondly, the learning loop can end if no uncertain unlabeled examples can be found in the pool. That is, all informative examples have been selected for annotation. However, this situation seldom occurs in real-world applications. Thirdly, the active learning process can stop if the targeted performance level is achieved. However, it is difficult to predefine an appropriate and achievable performance, since it should depend on the problem at hand and the users' requirements.

2.3 Problem of Performance Estimation

An appealing solution has the active learning process end when repeated cycles show no significant performance improvement on the test set. However, there are two open problems. The first question is how to measure the performance of a classifier in active learning. The second one is how to know when the resulting classifier reaches the highest or adequate performance. It seems feasible that a separate validation set can solve both problems. That is, the active learning process can end if there is no significant performance improvement on the validation set. But how many samples are required for the pregiven separate validation set is an open question. Too few samples may not be adequate for a reasonable estimation and may result in an incorrect result. Too many samples would cause additional high cost because the separate validation set is generally constructed manually in advance.

3 Statistical Learning Approach

3.1 Confidence-based Strategy

To avoid the problem of performance estimation mentioned above, Zhu and Hovy (2007) proposed a confidence-based framework to predict the upper bound and the lower bound for a stopping criterion in active learning. The motivation is to assume that the current training data is sufficient to train the classifier with maximum effectiveness if the current classifier already has acceptably strong confidence on its classification results for all remained unlabeled data.

The first method to estimate the confidence of the classifier is based on uncertainty measurement, considering whether the entropy of each selected unlabeled example is less than a small predefined threshold. Here we call it *Entropy-MCS*. The stopping criterion $SC_{Entropy\text{-}MCS}$ can be defined as:

$$SC_{Entropy-MCS} = \begin{cases} 1 & \forall x \in U, UM(x) \le \theta_E \\ 0 & otherwise, \end{cases} \quad (3)$$

where θ_E is a user predefined entropy threshold and the function *UM(x)* evaluates the uncertainty of each unlabeled example *x*.

The second method to estimate the confidence of the classifier is based on feedback from the oracle when the active learner asks for true labels for selected unlabeled examples, by considering whether the current trained classifier could correctly predict the labels or the accuracy performance of predictions on selected unlabeled examples is already larger than a predefined accuracy threshold. Here we call it *OracleAcc-MCS*. The stopping criterion $SC_{OracleAcc\text{-}MCS}$ can be defined as:

$$SC_{OracleAcc-MCS} = \begin{cases} 1 & OracleAcc(C) \ge \theta_A \\ 0 & otherwise, \end{cases} \quad (4)$$

where θ_A is a user predefined accuracy threshold and function *OracleAcc(C)* evaluates accuracy performance of the classifier on these selected unlabeled examples through feedback of the Oracle.

3.2 Minimum Expected Error Strategy

In fact, these above two confidence-based methods do not directly estimate classifier performance that closely reflects the classifier effectiveness, because they only consider entropy of each unlabeled example and accuracy on selected informative examples at each iteration step. In this section we therefore propose a new statistical learning approach to defining a stopping criterion through estimation of the classifier's expected error on all future unlabeled examples, which we call *minimum expected error strategy* (MES). The motivation behind MES is that the classifier *C* (a classifier for uncertainty sampling or set of classifiers for committee-based sampling) with maximum effectiveness is the one that results in the lowest expected

error on whole test set in the learning process. The stopping criterion SC_{MES} is defined as:

$$SC_{MES} = \begin{cases} 1 & Error(C) \le \theta_{err} \\ 0 & otherwise, \end{cases} \quad (5)$$

where θ_{err} is a user predefined expected error threshold and the function *Error(C)* evaluates the expected error of the classifier *C* that closely reflects the classifier effectiveness. So the key point of defining MES-based stopping criterion SC_{MES} is how to calculate the function *Error(C)* that denotes the expected error of the classifier *C*.

Suppose given a training set *L* and an input sample *x*, we can write the expected error of the classifier *C* as follows:

$$Error(C) = \int R(C(x) \mid x)P(x)dx \quad (6)$$

where *P(x)* represents the known marginal distribution of *x*. *C(x)* represents the classifier's decision that is one of *k* classes: $y \in Y=\{y_1, y_2, ..., y_k\}$. $R(y_i|x)$ denotes a conditional loss for classifying the input sample *x* into a class y_i that can be defined as

$$R(y_i \mid x) = \sum_{j=1}^{k} \lambda[i, j]P(y_j \mid x) \quad (7)$$

where $P(y_j|x)$ is the *a posteriori* probability produced by the classifier *C*. $\lambda[i,j]$ represents a zero-one loss function for every class pair $\{i,j\}$ that assigns no loss to a correct classification, and assigns a unit loss to any error.

In this paper, we focus on *pool-based active learning* in which a large unlabeled data pool *U* is available, as described Fig. 1. In active learning process, our interest is to estimate the classifier's expected error on future unlabeled examples in the pool *U*. That is, we can stop the active learning process when the active learner results in the lowest expected error over the unlabeled examples in *U*. The pool *U* can provide an estimate of *P(x)*. So for minimum error rate classification (Duda and Hart. 1973) on unlabeled examples, the expected error of the classifier *C* can be rewritten as

$$Error(C) = \frac{1}{|U|} \sum_{x \in U} (1 - \max_{y \in Y} P(y \mid x)) \quad (8)$$

Assuming *N* unlabeled examples in the pool *U*, the total time is *O(N)* for automatically determining whether the proposed stopping criterion SC_{MES} is satisfied in the active learning.

If the pool *U* is very large (e.g. more than 100000 examples), it would still cause high computation cost at each iteration of active learning. A good approximation is to estimate the expected error of the classifier using a subset of the pool, not using all unlabeled examples in *U*. In practice, a good estimation of expected error can be formed with few thousand examples.

4 Evaluation

In this section, we evaluate the effectiveness of three stopping criteria for active learning for word sense disambiguation and text classification as follows:

- *Entropy-MCS* — stopping active learning process when the stopping criterion function $SC_{Entropy\text{-}MCS}$ defined in (3) is equal to 1, where θ_E=*0.01, 0.001, 0.0001*.
- *OracleAcc-MCS* — stopping active learning process when the stopping criterion function $SC_{OracleAcc\text{-}MCS}$ defined in (4) is equal to 1, where θ_A=*0.9, 1.0*.
- *MES* — stopping active learning process when the stopping criterion function SC_{MES} defined in (5) is equal to 1, where θ_{err}=*0.01, 0.001, 0.0001*.

The purpose of defining stopping criterion of active learning is to study how much annotation is sufficient for a specific task. To comparatively analyze the effectiveness of each stopping criterion, a *baseline* stopping criterion is predefined as when all unlabeled examples in the pool *U* are learned. Comparing with the baseline stopping criterion, a better stopping criterion not only achieves almost the same performance, but also has needed to learn fewer unlabeled examples when the active learning process is ended. In other words, for a stopping criterion of active learning, the fewer unlabeled examples that have been leaned when it is met, the bigger reduction in human labeling cost is made.

In the following active learning experiments, a 10 by 10-fold cross-validation was performed. All results reported are the average of 10 trials in each active learning process.

4.1 Word Sense Disambiguation

The first comparison experiment is active learning for word sense disambiguation. We utilize a maximum entropy (ME) model (Berger *et al.*, 1996) to design the basic classifier used in active learning for WSD. The advantage of the ME model is the ability to freely incorporate features from

diverse sources into a single, well-grounded statistical model. A publicly available ME toolkit (Zhang *et. al.*, 2004) was used in our experiments. In order to extract the linguistic features necessary for the ME model in WSD tasks, all sentences containing the target word are automatically part-of-speech (POS) tagged using the Brill POS tagger (Brill, 1992). Three knowledge sources are used to capture contextual information: unordered single words in topical context, POS of neighboring words with position information, and local collocations. These are same as the knowledge sources used in (Lee and Ng, 2002) for supervised automated WSD tasks.

The data used for comparison experiments was developed as part of the OntoNotes project (Hovy *et al*., 2006), which uses the WSJ part of the Penn Treebank (Marcus *et al.*, 1993). The senses of noun words occurring in OntoNotes are linked to the Omega ontology (philpot *et al.*, 2005). In OntoNotes, at least two human annotators manually annotate the coarse-grained senses of selected nouns and verbs in their natural sentence context. In this experiment, we used several tens of thousands of annotated OntoNotes examples, covering in total 421 nouns with an inter-annotator agreement rate of at least 90%. We find that 302 out of 421 nouns occurring in OntoNotes are ambiguous, and thus are used in the following WSD experiments. For these 302 ambiguous nouns, there are 3.2 senses per noun, and 172 instances per noun.

The active learning algorithms start with a randomly chosen initial training set of 10 labeled samples for each noun, and make 10 queries after each learning iteration. Table 1 shows the effectiveness of each stopping criterion tested on active learning for WSD on these ambiguous nouns' WSD tasks. We analyze average accuracy performance of the classifier and average percentage of unlabeled examples learned when each stopping criterion is satisfied in active learning for WSD tasks. All accuracies and percentages reported in Table 1 are macro-averages over these 302 ambiguous nouns.

Stopping Criterion	Average accuracy	Average percentage
all unlabeled examples learned	87.3%	100%
Entropy-MCS method (0.0001)	86.8%	81.8%
Entropy-MCS method (0.001)	86.8%	75.8%
Entropy-MCS method (0.01)	86.8%	68.6%
OracleAcc-MCS method (0.9)	86.8%	56.5%
OracleAcc-MCS method (1.0)	86.8%	62.4%
MES method (0.0001)	86.8%	67.1%
MES method (0.001)	86.8%	58.8%
MES method (0.01)	86.8%	52.7%

Table 1. Effectiveness of each stopping criterion of active learning for WSD on OnteNotes.

Table 1 shows that these stopping criteria achieve the same accuracy of 86.8% which is within 0.5% of the accuracy of the baseline method (all unlabeled examples are labeled). It is obvious that these stopping criteria can help reduce the human labeling costs, comparing with the baseline method. The best criterion is MES method (θ_{err}=0.01), following by OracleAcc-MCS method (θ_A=0.9). MES method (θ_{err}=0.01) and OracleAcc-MCS method (θ_A=0.9) can make 47.3% and 44.5% reductions in labeling costs, respectively. Entropy-MCS method is apparently worse than MES and OracleAcc-MCS methods. The best of the Entropy-MCS method is the one with θ_E=0.01 which makes approximately 1/3 reduction in labeling costs. We also can see from Table 1 that for Entropy-MCS and MES methods, reduction rate becomes smaller as the θ becomes smaller.

4.2 Text Classification

The second data set is for active learning for text classification using the WebKB corpus [1] (McCallum *et al.*, 1998). The WebKB dataset was formed by web pages gathered from various university computer science departments. In the following active learning experiment, we use four most populous categories: *student, faculty, course* and *project*, altogether containing 4,199 web pages. Following previous studies (McCallum *et al.*, 1998), we only remove those words that occur merely once without using stemming or stop-list. The resulting vocabulary has 23,803 words. In the design of the text classifier, the maximum entropy model is also utilized, and no feature selection technique is used.

[1] See http://www.cs.cmu.edu/~textlearning

The algorithm is initially given 20 labeled examples, 5 from each class. Table 2 shows the effectiveness of each stopping criterion of active learning for text classification on WebKB corpus. All results reported are the average of 10 trials.

Stopping Criterion	Average accuracy	Average percentage
all unlabeled examples learned	93.5%	100%
Entropy-MCS method (0.0001)	92.5%	23.8%
Entropy-MCS method (0.001)	92.4%	22.3%
Entropy-MCS method (0.01)	92.5%	21.8%
OracleAcc-MCS method (0.9)	91.5%	13.1%
OracleAcc-MCS method (1.0)	92.5%	24.5%
MES method (0.0001)	92.1%	17.9%
MES method (0.001)	92.0%	15.6%
MES method (0.01)	91.5%	10.9%

Table 2. Effectiveness of each stopping criterion of active learning for TC on WebKB corpus.

From results shown in Table 2, we can see that MES method (θ_{err}=0.01) already achieves 91.5% accuracy in 10.9% unlabeled examples learned. The accuracy of all unlabeled examples learned is 93.5%. This situation means the approximately 90% remaining unlabeled examples only make only 2% performance improvement. Like the results of WSD shown in Table 1, for Entropy-MCS and MES methods used in active learning for text classification tasks, the corresponding reduction rate becomes smaller as the value of θ becomes smaller. MES method (θ_{err}=0.01) can make approximately 90% reduction in human labeling costs and results in 2% accuracy performance degradation. The Entropy-MCS method (θ_E=0.01) can make approximate 80% reduction in costs and results in 1% accuracy performance degradation. Unlike the results of WSD shown in Table 1, the OracleAcc-MCS method (θ_A=1.0) makes the smallest reduction rate of 75.5%. Actually in real-world applications, the selection of a stopping criterion is a tradeoff issue between labeling cost and effectiveness of the classifier.

5 Discussion

It is interesting to investigate the impact of performance change on defining a stopping criterion, so we show an example of active learning for WSD task in Fig. 2.

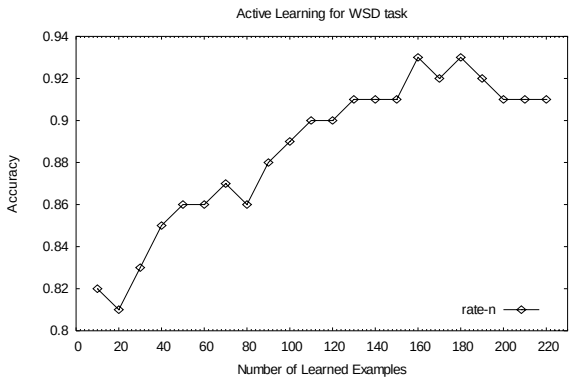


Figure 2. An example of active learning for WSD on noun "*rate*" in OntoNotes.

Fig. 2 shows that the accuracy performance generally increases, but apparently degrades at the iterations "*20*", "*80*", "*170*", "*190*", and "*200*", and does not change anymore during the iterations ["*130*"-"*150*"] or ["*200*"-"*220*"] in the active learning process. Actually the first time of the highest performance of 95% achieved is at "*450*", which is not shown in Fig. 2. In other words, although the accuracy performance curve shows an increasing trend, it is not monotonously increasing. From Fig. 2 we can see that it is not easy to automatically determine the point of no significant performance improvement on the validation set, because points such as "*20*" or "*80*" would mislead final judgment. However, we do believe that the change of performance is a good signal to stop active learning process. So it is worth studying further how to combine the factor of performance change with our proposed stopping criteria of active learning.

The OracleAcc-MCS method would not work if only one or too few informative examples are queried at the each iteration step in the active learning. There is an open issue how many selected unlabeled examples at each iteration are adequate for the batch-based sample selection.

For these stopping crieria, there is no general method to automatically determine the best threshold for any given task. It may therefore be necessary to use a dynamic threshold change technique in which the predefined threshold can be automatically modified if the performance is still significantly improving when the stopping criterion is met during active learning process.

6 Conclusion and Future Work

In this paper, we address the stopping criterion issue of active learning, and analyze the problems faced by some common ways to stop the active learning process. In essence, defining a stopping criterion of active learning is a problem of estimating classifier effectiveness. The purpose of defining stopping criterion of active learning is to know how much annotation is sufficient for a special task. To determine this, this paper proposes a new statistical learning approach, called minimum expected error strategy, for defining a stopping criterion through estimation of the classifier's expected error on future unlabeled examples during the active learning process. Experimental results on word sense disambiguation and text classification tasks show that new proposed minimum expected error strategy outperforms the confidence-based strategy, and achieves promising results. The interesting future work is to study how to combine the best of both strategies, and how to consider performance change to define an appropriate stopping criterion for active learning.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant (60473140), the National 863 High-tech Project (2006AA01Z154); the Program for New Century Excellent Talents in University(NCET-05-0287).

References

A. L. Berger, S. A. Della, and V. J Della. 1996. *A maximum entropy approach to natural language processing*. Computational Linguistics 22(1):39–71.

E Brill. 1992. *A simple rule-based part of speech tagger*. In the Proceedings of the Third Conference on Applied Natural Language Processing.

J. Chen, A. Schein, L. Ungar, M. Palmer. 2006. *An empirical study of the behavior of active learning for word sense disambiguation*. In Proc. of HLT-NAACL06

R. O. Duda and P. E. Hart. 1973. *Pattern classification and scene analysis*. New York: Wiley.

S. A. Engelson and I. Dagan. 1999. *Committee-based sample selection for probabilistic classifiers*. Journal of Artificial Intelligence Research.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw and R. Weischedel. 2006. *Ontonotes: The 90% Solution*. In Proc. of HLT-NAACL06.

Y.K. Lee and. H.T. Ng. 2002. *An empirical evaluation of knowledge sources and learning algorithm for word sense disambiguation*. In Proc. of EMNLP02

D. D. Lewis and W. A. Gale. 1994. *A sequential algorithm for training text classifiers*. In Proc. of SIGIR-94

M. Li, I. K. Sethi. 2006. *Confidence-based active learning*. IEEE transaction on pattern analysis and machine intelligence, 28(8):1251-1261.

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. *Building a large annotated corpus of English: the Penn Treebank*. Computational Linguistics, 19(2):313-330

A. McCallum and K. Nigram. 1998. *Employing EM in pool-based active learning for text classification*. In Proc. of 15th ICML

G. Ngai and D. Yarowsky. 2000. *Rule writing or annotation: cost-efficient resource usage for based noun phrase chunking*. In Proc. of ACL-02

A. Philpot, E. Hovy and P. Pantel. 2005. *The Omega Ontology*. In Proc. of ONTOLEX Workshop at IJCNLP.

D. Shen, J. Zhang, J. Su, G. Zhou and C. Tan. 2004. *Multi-criteria-based active learning for named entity recognition*. In Prof. of ACL-04.

M. Steedman, R. Hwa, S. Clark, M. Osborne, A. Sakar, J. Hockenmaier, P. Ruhlen, S. Baker and J. Crim. 2003. *Example selection for bootstrapping statistical parsers*. In Proc. of HLT-NAACL-03

C. A. Thompson, M. E. Califf and R. J. Mooney. 1999. *Active learning for natural language parsing and information extraction*. In Proc. of ICML-99.

K. Tomanek, J. Wermter and U. Hahn. 2007. *An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data*. In Proc. of EMNLP/CoNLL07

L. Zhang, J. Zhu, and T. Yao. 2004. *An evaluation of statistical spam filtering techniques*. ACM Transactions on Asian Language Information Processing, 3(4):243–269.

J. Zhu, E. Hovy. 2007. *Active learning for word sense disambiguation with methods for addressing the class imbalance problem*. In Proc. of EMNLP/CoNLL07

Multi-View Co-training of Transliteration Model

Jin-Shea Kuo
Chung-Hwa Telecomm. Laboratories, Taiwan
d8807302@gmail.com

Haizhou Li
Institute for Infocomm Research, Singapore 119613
hli@i2r.a-star.edu.sg

Abstract

This paper discusses a new approach to training of transliteration model from unlabeled data for transliteration extraction. We start with an inquiry into the formulation of transliteration model by considering different transliteration strategies as a multi-view problem, where each view exploits a natural division of transliteration features, such as phoneme-based, grapheme-based or hybrid features. Then we introduce a multi-view Co-training algorithm, which leverages *compatible* and partially *uncorrelated* information across different views to effectively boost the model from unlabeled data. Applying this algorithm to transliteration extraction, the results show that it not only circumvents the need of data labeling, but also achieves performance close to that of supervised learning, where manual labeling is required for all training samples.

1 Introduction

Named entities are important content words in text documents. In many applications, such as cross-language information retrieval (Meng et al., 2001; Virga and Khudanpur, 2003) and machine translation (Knight and Graehl, 1998; Chen et al., 2006), one of the fundamental tasks is to identify these words. Imported foreign proper names constitute a good portion of such words, which are newly translated into Chinese by transliteration. Transliteration is a process of translating a foreign word into the native language by preserving its pronunciation in the original language, otherwise known as *translation-by-sound*.

As new words emerge everyday, no lexicon is able to cover all transliterations. It is desirable to find ways to harvest transliterations from real world corpora. In this paper, we are interested in the learning of English to Chinese (*E-C*) transliteration model for transliteration extraction from the Web.

A statistical transliteration model is typically trained on a large amount of transliteration pairs, also referred to a bilingual corpus. The correspondence between a transliteration pair may be described by the mapping of different basic pronunciation units (BPUs) such as phoneme-based[1], or grapheme-based one, or both. We can see each type of BPU mapping as a natural division of transliteration features, which represents a view to the phonetic mapping problem. By using different BPUs, we approach the transliteration modeling and extraction problems from different views.

This paper is organized as follows. In Section 2, we briefly introduce previous work. In Section 3, we conduct an inquiry into the formulation of transliteration model or phonetic similarity model (PSM) and consider it as a multi-view problem. In Section 4, we propose a multi-view Co-training strategy for PSM training and transliteration extraction. In Section 5, we study the effectiveness of proposed algorithms. Finally, we conclude in Section 6.

2 Related Work

Studies on transliteration have been focused on transliteration modeling and transliteration extraction. The transliteration modeling approach deduces either phoneme-based or grapheme-based mapping rules using a generative model that is

[1] Both phoneme and syllable based approaches are referred to as phoneme-based in this paper.

trained from a large bilingual corpus. Most of the works are devoted to phoneme-based transliteration modeling (Knight and Graehl, 1998; Lee, 1999). Suppose that *EW* is an English word and *CW* is its Chinese transliteration. *EW* and *CW* form an *E-C* transliteration pair. The phoneme-based approach first converts *EW* into an intermediate phonemic representation *p*, and then converts *p* into its Chinese counterpart *CW*. The idea is to transform both source and target words into comparable phonemes so that the phonetic similarity between two words can be measured easily.

Recently the grapheme-based approach has attracted much attention. It was proposed by Jeong et al. (1999), Li et al. (2004) and many others (Oh et al., 2006b), which is also known as direct orthography mapping. It treats the transliteration as a statistical machine translation problem under monotonic constraint. The idea is to obtain the bilingual orthographical correspondence directly to reduce the possible errors introduced in multiple conversions. However, the grapheme-based transliteration model has more parameters than phoneme-based one does, thus expects a larger training corpus.

Most of the reported works have been focused on either phoneme- or grapheme-based approaches. Bilac and Tanaka (2004) and Oh et al. (2006a; 2006b) recently proposed using a mix of phoneme and grapheme features, where both features are fused into a single learning process. The feature fusion was shown to be effective. However, their methods hinge on the availability of a labeled bilingual corpus.

In transliteration extraction, mining translations or transliterations from the ever-growing multilingual Web has become an active research topic, for example, by exploring query logs (Brill et al., 2001) and parallel (Nie et al., 1999) or comparable corpora (Sproat et al., 2006). Transliterations in such a live corpus are typically unlabeled. For model-based transliteration extraction, recent progress in machine learning offers different options to exploit unlabeled data, that include active learning (Lewis and Catlett, 1994) and Co-training (Nigam and Ghani, 2000; Tür et al. 2005).

Taking the prior work a step forward, this paper explores a new way of fusing phoneme and grapheme features through a multi-view Co-training algorithm (Blum and Mitchell, 1998), which starts with a small number of labeled data to bootstrap a transliteration model to automatically harvest transliterations from the Web.

3 Phonetic Similarity Model with Multiple Views

Machine transliteration can be formulated as a generative process, which takes a character string in source language as input and generates a character string in the target language as output. Conceptually, this process can be regarded as a 3-step decoding: segmentation of both source and target strings into basic pronunciation units (BPUs), relating the source BPUs with target units by resolving different combinations of alignments and unit mappings in finding the most probable BPU pairs. A BPU can be defined as a phoneme sequence, a grapheme sequence, or a part of them. A transliteration model establishes the phonetic relationship between BPUs in two languages to measure their similarity, therefore, it is also known as the phonetic similarity model (PSM).

To introduce the multi-view concept, we illustrate the BPU transfers in Figure 1, where each transfer is represented by a direct path with different line style. There are altogether four different paths: the phoneme-based path V_1 ($T_1 \rightarrow T_2 \rightarrow T_3$), the grapheme-based path V_4 (T_4), and their variants, V_2($T_1 \rightarrow T_5$) and V_3($T_6 \rightarrow T_3$). The last two paths make use of the intermediate BPU mappings between phonemes and graphemes. Each of the paths represents a view to the mapping problem. Given a labeled bilingual corpus, we are able to train a transliteration model for each view easily.

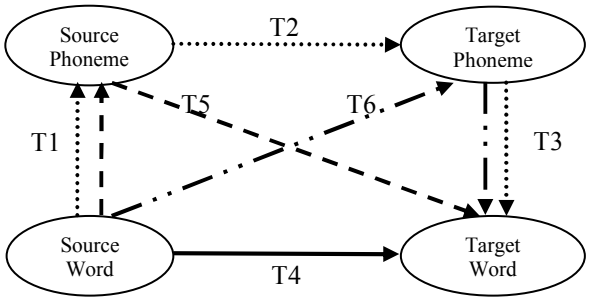


Figure 1. Multiple views for establishing transliteration correspondence.

The *E-C* transliteration has been studied extensively in the paradigm of noisy channel model

(Manning and Scheutze, 1999), with *EW* as the observation and *CW* as the input to be recovered. Applying Bayes rule, the transliteration can be described by Eq. (1),

$$P(CW \mid EW) = \frac{P(EW \mid CW) \times P(CW)}{P(EW)}, \quad (1)$$

where we need to deal with two probability distributions: *P(EW|CW)*, the probability of transliterating *CW* to *EW*, also known as the unit mapping rules, and *P(CW)*, the probability distribution of *CW*, known as the target language model.

Representing *EW* in English BPU sequence $EP = \{ep_1, ...ep_m, ...ep_M\}$ and *CW* in Chinese one, $CP = \{cp_1, ...cp_n, ...cp_N\}$, a typical transliteration probability can be expressed as,

$$P(EW \mid CW) \approx P(EW \mid EP) \times P(EP \mid CP) \times P(CP \mid CW). \quad (2)$$

The language model, *P(CW)*, can be represented by Chinese characters *n*-gram statistics (Manning and Scheutze, 1999) and expressed in Eq. (3). In the case of bigram, we have,

$$P(CW) \approx P(c_1) \prod_{n=2}^{N} P(c_n \mid c_{n-1}) \quad (3)$$

We next rewrite Eq. (2) for the four different views depicted in Figure 1 in a systematic manner.

3.1 Phoneme-based Approach

The phoneme-based approach approximates the transliteration probability distribution by introducing an intermediate phonemic representation. In this way, we convert the words in the source language, say $EW = e_1, e_2 ... e_K$, into English syllables *ES*, then Chinese syllables *CS* and finally the target language, say Chinese $CW = c_1, c_2 ... c_K$ in sequence. Eq. (2) can be rewritten by replacing *EP* and *CP* with *ES* and *CS*, respectively, and expressed by Eq. (4).

$$P(EW \mid CW) \approx P(EW \mid ES) \times P(ES \mid CS) \times P(CS \mid CW) \quad (4)$$

The three probabilities correspond to the three-step mapping in V_1 path.

The phoneme-based approach suffers from multiple step mappings. This could compromise overall performance because none of the three steps guarantees a perfect conversion.

3.2 Grapheme-based Approach

The grapheme-based approach is inspired by the transfer model (Vauqois, 1988) in machine translation that estimates $P(EW \mid CW)$ directly without interlingua representation. This method aims to alleviate the imprecision introduced by the multiple transfers in phoneme-based approach.

In practice, a grapheme-based approach converts the English graphemes to Chinese graphemes in one single step. Suppose that we have $EW = e_1, e_2 ... e_K$ and $CW = c_1, c_2 ... c_K$ where e_k and c_k are aligned grapheme units.

Under the noisy channel model, we can estimate $P(EW \mid CW)$ based on the alignment statistics which is similar to the lexical mapping in statistical machine translation.

$$P(EW \mid CW) \approx \prod_{k=1}^{K} P(e_k \mid c_k) \quad (5)$$

Eq.(5) is a grapheme-based alternative to Eq.(2).

3.3 Hybrid Approach

A tradeoff between the phoneme- and grapheme-based approaches is to take shortcuts to the mapping between phonemes and graphemes of two languages via V_2 or V_3, where only two steps of mapping are involved. For V_3, we rewrite Eq.(2) as Eq. (6):

$$P(EW \mid CW) = P(EW \mid CS) \times P(CS \mid CW), \quad (6)$$

where $P(EW \mid CS)$ translates Chinese sounds into English words. For V_2, we rewrite Eq. (2) as Eq. (7):

$$P(EW \mid CW) = P(EW \mid ES) \times P(ES \mid CW), \quad (7)$$

where $P(ES \mid CW)$ translates Chinese words into English sounds.

Eqs. (4) – (7) describe the four paths of transliteration. In a multi-view problem, one partitions the domain's features into subsets, each of which is sufficient for learning the target concept. Here the target concept is the label of transliteration pair. Given a collection of *E-C* pair candidates, the transliteration extraction task can be formulated as a hypothesis test, which makes a binary decision as to whether a candidate *E-C* pair is a genuine transliteration pair or not. Given an *E-C* pair $X=\{EW,CW\}$, we have H_0, which

hypothesizes that *EW* and *CW* form a genuine *E-C* pair, and H_1, which hypothesizes otherwise. The likelihood ratio is given as $\sigma = P(X \mid H_0) / P(X \mid H_1)$, where $P(X \mid H_0)$ and $P(X \mid H_0)$ are derived from *P(EW|CW)*. By comparing σ with a threshold τ, we make the binary decision as that in (Kuo et al., 2007).

As discussed, each view takes a distinct path that has its own advantages and disadvantages in terms of model expressiveness and complexity. Each view represents a weak learner achieving moderately good performance towards the target concept. Next, we study a multi-view Co-training process that leverages the data of different views from each other in order to boost the accuracy of a PSM model.

4 Multi-View Learning Framework

The PSM can be trained in a supervised manner using a manually labeled corpus. The advantage of supervised learning is that we can establish a model quickly as long as labeled data are available. However, this method suffers from some practical constraints. First, the derived model can only be as good as the data it sees. Second, the labeling of corpus is labor intensive.

To circumvent the need of manual labeling, here we study three adaptive strategies cast in the machine learning framework, namely unsupervised learning, Co-training and Co-EM.

4.1 Unsupervised Learning

Unsupervised learning minimizes human supervision by probabilistically labeling data through an Expectation and Maximization (EM) (Dempster et al., 1977) process. The unsupervised learning strategy can be depicted in Figure 2 by taking the dotted path, where the extraction process accumulates all the acquired transliteration pairs in a repository for training a new PSM. A new PSM is in turn used to extract new transliteration pairs. The unsupervised learning approach only needs a few labeled samples to bootstrap the initial model for further extraction. Note that the training samples are noisy and hence the quality of initial PSM therefore has a direct impact on the final performance.

4.2 Co-training and Co-EM

The multi-view setting (Muslea et al., 2002) applies to learning problems that have a natural way to divide their features into different views, each of which is sufficient to learn the target concept. Blum and Mitchell (1998) proved that for a problem with two views, the target concept can be learned based on a few labeled and many unlabeled examples, provided that the views are *compatible* and *uncorrelated*. Intuitively, the transliteration problem has *compatible* views. If an *E-C* pair forms a transliteration, then this is true across all different views. However, it is arguable that the four views in Figure 1 are *uncorrelated*. Studies (Nigam and Ghani, 2000; Muslea et al., 2002) shown that the views do not have to be entirely *uncorrelated* for Co-training to take effect. This motivates our attempt to explore multi-view Co-training for learning models in transliteration extraction.

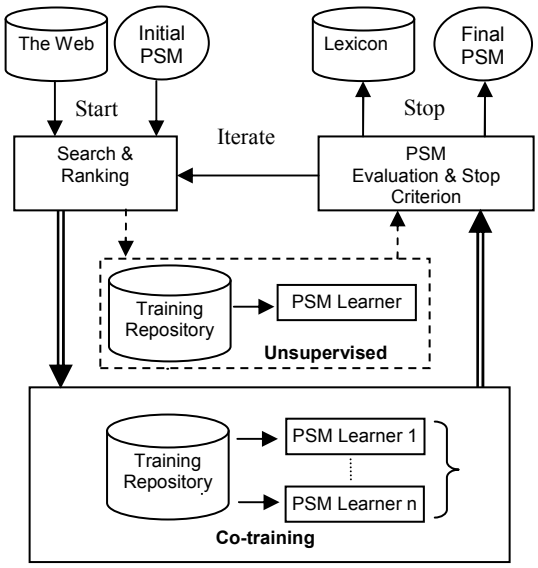


Figure 2. Diagram of unsupervised/multi-view Co-training for transliteration extraction.

To simplify the discussion, here we take a two-view (V_1 and V_2) example to show how Co-training can potentially help. To start with, one can learn a weak hypothesis PSM_1 using V_1 based on a few labeled examples and then apply PSM_1 to all unlabeled examples. If the views are *uncorrelated*, or at least partially *uncorrelated,* these newly labeled examples seen from V_1 augment the training set for V_2. These newly labeled examples

present new information from the V_2 point of view, from which one can in turn update the PSM_2. As the views are *compatible,* both V_1 and V_2 label the samples consistently according to the same probabilistic transliteration criteria. In this way, PSMs are boosted each other through such an iterative process between two different views.

Given:
- a). A small set of labeled samples and a set of unlabeled samples.
- b). Two learners A and B are trained on the labeled set.

1) Loop for *k* iterations:
- a). Learners A and B predict the labels of the unlabeled data to augment the labeled set;
- b). Learners A and B are trained on the augmented labeled set.

2) Combine models from Learners A and B.

Table 1. Co-training with two learners.

Extending the two-view to multi-view, one can develop multiple learners from several subsets of features, each of which approaches the problem from a unique perspective, called a view when taking the Co-training path in Figure 2. Finally, we use outputs from multi-view learners to approximate the manual labeling. The multi-view learning is similar to unsupervised learning in the sense that the learning alleviates the need of labeling and starts with very few labeled data. However, it is also different from the unsupervised learning because the latter does not leverage the natural split of *compatible* and *uncorrelated* features. Two variants of two-view learning strategy can be summarized in Table 1 and Table 2, where the algorithm in Table 1 is referred to as Co-training and the one in Table 2 as Co-EM (Nigam and Ghani. 2000; Muslea et al., 2002).

In Co-training, Learners A and B are trained on the same training data and updated simultaneously. In Co-EM, Learners A and B are trained on labeled set predicted by each other's view, with their models being updated in sequence. In other words, the Co-EM algorithm interchanges the probabilistic labels generated in the view of each other before a new EM iteration. In both cases, the unsupervised, multi-view algorithms use the hypotheses learned to probabilistically label the examples.

Given
- a). A small set of labeled samples and a set of unlabeled samples.
- b). Learner A is trained on a labeled set to predict the labels of the unlabeled data.

1) Loop for *k* iterations
- a). Learner B is trained on data labeled by Learner A to predict the labels of the unlabeled data;
- b). Learner A is trained on data labeled by Learner B to predict the labels of the unlabeled data;

2) Combine models from Learners A and B.

Table 2. Co-EM with two learners.

The extension of algorithms in Table 1 and 2 to the multi-view transliteration problem is straightforward. After an ensemble of learners are trained, the overall PSM can be expressed as a linear combination of the learners,

$$P(EW \mid CW) = \sum_{i=1}^{n} w_i P_i(EW \mid CW), \quad (8)$$

where w_i is the weight of i^{th} learner $P_i(EW \mid CW)$, which can be learnt by using a development corpus.

5 Experiments

To validate the effectiveness of the learning framework, we conduct a series of experiments in transliteration extraction on a development corpus described later. First, we repeat the experiment in (Kuo et al., 2006) to train a PSM using PSA and GSA feature fusion in a supervised manner, which serves as the upper bound of Co-training or Co-EM system performance. We then train the PSMs with single view V1, V2, V3 and V4 alone in an unsupervised manner. The performance achieved by each view alone can be considered as the baseline for multi-view benchmarking. Then, we run two-view Co-training for different combinations of views on the same development corpus. We expect to see positive effects with the multi-view training. Finally, we run the experiments using two-view Co-training and Co-EM and compare the results.

A 500 MB development corpus is constructed by crawling pages from the Web for the experiments. We first establish a gold standard for performance evaluation by manually labeling the corpus based on the following criteria: (i) if an *EW* is partly

translated phonetically and partly translated semantically, only the phonetic transliteration constituent is extracted to form a transliteration pair; (ii) multiple *E-C* pairs can appear in one sentence; (iii) an *EW* can have multiple valid Chinese transliterations and vice versa.

We first derive 80,094 *E-C* pair candidates from the 500 MB corpus by spotting the co-occurrence of English and Chinese words in the same sentences. This can be done automatically without human intervention. Then, the manual labeling process results in 8,898 qualified *E-C* pairs, also referred to as Distinct Qualified Transliteration Pairs (DQTPs).

To establish comparison, we first train a PSM using all 8,898 DQTPs in a supervised manner and conduct a closed test as reported in Table 3. We further implement three PSM learning strategies and conduct a systematic series of experiments by following the *recognition followed by validation* strategy proposed in (Kuo et al., 2007).

	Precision	Recall	F-measure
Closed test	0.834	0.663	0.739

Table 3. Performance with PSM trained in the supervised manner.

For performance benchmarking, we define the *precision* as the ratio of extracted number of DQTPs over that of total extracted pairs, *recall* as the ratio of extracted number of DQTPs over that of total DQTPs, and *F*-measure as in Eq. (9). They are collectively referred to as extraction performance.

$$F-measure = \frac{2 \times recall \times precision}{recall + precision} \quad (9)$$

5.1 Unsupervised Learning

As formulated in Section 4.1, first, we derive an initial PSM using randomly selected 100 seed DQTPs for each learner and simulate the Web-based learning process: (i) extract *E-C* pairs using the PSM; (ii) add all of the extracted *E-C* pairs to the DQTP pool; (iii) re-estimate the PSM for each view by using the updated DQTP pool. This process is also known as semi-supervised EM (Muslea et al., 2002).

As shown in Figure 3, the unsupervised learning algorithm consistently improves the initial PSM using in all four views. To appreciate the effectiveness of each view, we report the F-measures on each individual view V_1, V_2, V_3 and V_4, as 0.680, 0.620, 0.541 and 0.520, respectively at the 6th iteration. We observe that V_1, the phoneme-based path, achieves the best result.

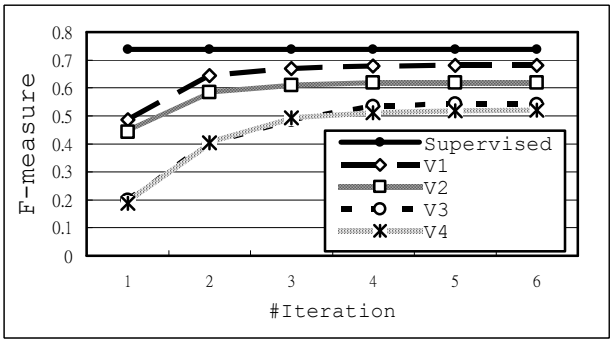


Figure 3. F-measure over iterations using unsupervised learning with individual view.

5.2 Co-training (CT)

We report three typical combinations of two co-working learners or two-view Co-training. Like in unsupervised learning, we start with the same 100 seed DQTPs and an initial PSM model by following the algorithm in Table 1 over 6 iterations.

With two-view Co-training, we obtain 0.726, 0.705, 0.590 and 0.716 in terms of F-measures for V_1+V_2, V_2+V_3, V_3+V_4 and V_1+V_4 at the 6th iteration, as shown in Figure 4. Comparing Figure 3 and 4, we find that Co-training consistently outperforms unsupervised learning by exploiting *compatible* information across different views. The V_1+V_2 Co-training outperforms other Co-training combinations, and surprisingly achieves close performance to that of supervised learning.

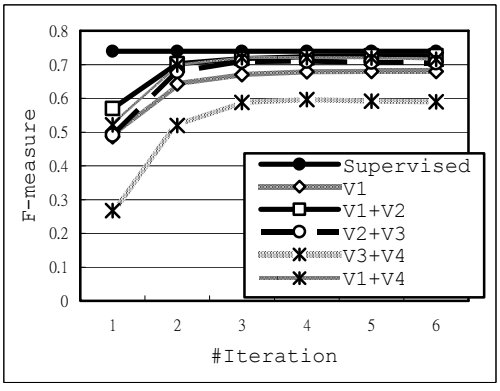


Figure 4. F-measure over iterations using Co-training algorithm

5.3 Co-EM (CE)

Next we start with the same 100 seed DQTPs by initializing the training pool and carry out Co-EM on the same corpus. We build PSM_1 for Learner A and PSM_2 for Learner B. To start with, PSM_1 is learnt from the initial labeled set. We then follow the algorithm in Table 2 by looping in the following two steps over 6 iterations: (i) estimate the PSM_2 from the samples labeled by Learner A (V_1) to extract the high confident *E-C* pairs and augment the DQTP pool with the probabilistically labeled *E-C* pairs; (ii) estimate the PSM_1 from the samples labeled by Learner B (V_2) to extract the high confident *E-C* pairs and augment the DQTP pool with the probabilistically labeled *E-C* pairs. We report the results in Figure 5.

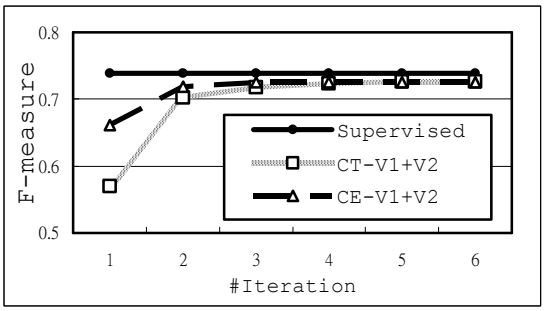


Figure 5. Comparing F-measure over iterations between Co-training (CT) and Co-EM (CE).

To summarize, we compare the performance of six learning methods studied in this paper in Table 4. The Co-training and Co-EM learning approaches have alleviated the need of manual labeling, yet achieving performance close to supervised learning. The multi-view learning effectively leverages multiple *compatible* and partially *uncorrelated* views. It reduces the need of labeled samples from 80,094 to just 100.

We also compare the multi-view learning algorithm with active learning on the same development corpus using same features. We include the results from previously reported work (Kuo et al., 2006) into Table 4 (see Exp. 2) where multiple features are fused in a single active learning process. In Exp. 2, PSA feature is the equivalent of V1 feature in Exp. 4; GSA feature is the equivalent of V4 feature in Exp. 4. In Exp. 4, we carry out V1+V4 two-view Co-training. It is interesting to find that the multi-view learning in this paper achieves better results than active learning in terms of F-measure while reducing the need of manual labeling from 8,191 samples to just 100.

Exp.	Learning algorithm	F-measure	# of samples to label
1	Supervised	0.739	80,094
2	Active Learning (Kuo et al., 2006)	0.710	8,191
3	Unsupervised (V_1)	0.680	100
4	Co-training (V_1+V_4)	0.716	100
5	Co-training (V_1+V_2)	0.726	100
6	Co-EM (V_1+V_2)	0.725	100

Table 4. Comparison of six learning strategies.

6 Conclusions

Fusion of phoneme and grapheme features in transliteration modeling was studied in many previous works. However, it was done through the combination of phoneme and grapheme similarity scores (Bilac and Tanaka, 2004), or by pooling phoneme and grapheme features together into a single-view training process (Oh and Choi, 2006b). This paper presents a new approach that leverages the information across different views to effectively boost the learning from unlabeled data.

We have shown that both Co-training and Co-EM not only outperform the unsupervised learning of single view, but also alleviate the need of data labeling. This reaffirms that multi-view is a viable solution to the learning of transliteration model and hence transliteration extraction. Moving forward, we believe that contextual feature in documents presents another *compatible*, *uncorrelated*, and complementary view to the four views.

We validate the effectiveness of the proposed algorithms by conducting experiments on transliteration extraction. We hope to extend the work further by investigating the possibility of applying the multi-view learning algorithms to machine translation.

References

S. Bilac and H. Tanaka. 2004. Improving back-transliteration by combining information sources, In *Proc. of Int'l Joint Conf. on Natural Language Processing,* pp. 542-547.

S. Blum and T. Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-training, In *Proc. of 11th Conference on Computational Learning Theory*, pp. 92-100.

E. Brill, G. Kacmarcik and C. Brockett. 2001. Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs, In *Proc. of Natural Language Processing Pacific Rim Symposium (NLPPRS)*, pp. 393-399.

H.-H. Chen, W.-C. Lin, C.-H. Yang and W.-H. Lin. 2006, Translating-Transliterating Named Entities for Multilingual Information Access, Journal of the American Society for Information Science and Technology, 57(5), pp. 645-659.

A. P. Dempster, N. M. Laird and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society, Ser. B. Vol. 39, pp. 1-38.

K. S. Jeong, S. H. Myaeng, J. S. Lee and K.-S. Choi. 1999. Automatic Identification and Back-transliteration of Foreign Words for Information Retrieval, Information Processing and Management, Vol. 35, pp. 523-540.

K. Knight and J. Graehl. 1998. Machine Transliteration, Computational Linguistics, Vol. 24, No. 4, pp. 599-612.

J.-S. Kuo, H. Li and Y.-K. Yang. 2006. Learning Transliteration Lexicons from the Web, In *Proc. of 44th ACL*, pp. 1129-1136.

J.-S. Kuo, H. Li and Y.-K. Yang. 2007. A Phonetic Similarity Model for Automatic Extraction of Transliteration Pairs, ACM Transactions on Asian Language Information Processing. 6(2), pp. 1-24.

J.-S. Lee. 1999. An English-Korean Transliteration and Retransliteration Model for Cross-Lingual Information Retrieval, PhD Thesis, Department of Computer Science, KAIST.

D. D. Lewis and J. Catlett. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning, In *Proc. of Int'l Conference on Machine Learning (ICML)*, pp. 148-156.

H. Li, M. Zhang and J. Su. 2004. A Joint Source Channel Model for Machine Transliteration, In *Proc. of 42nd ACL*, pp. 159-166.

C. D. Manning and H. Scheutze. 1999. Fundamentals of Statistical Natural Language Processing, The MIT Press.

H. M. Meng, W.-K. Lo, B. Chen and T. Tang. 2001. Generate Phonetic Cognates to Handle Name Entities in English-Chinese Cross-Language Spoken Document Retrieval, In *Proceedings of Automatic Speech Recognition Understanding (ASRU)*, pp. 311-314.

I. Muslea, S. Minton and C. A. Knoblock. 2002. Active + Semi-supervised learning = Robust Multi-View Learning, In *Proc. of the 9th Int'l Conference on Machine Learning*, pp. 435-442.

J.-Y. Nie, P. Isabelle, M. Simard and R. Durand. 1999. Cross-language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Text from the Web, In *Proc. of 22nd ACM SIGIR*, pp 74-81.

K. Nigam and R. Ghani. 2000. Analyzing the Effectiveness and Applicability of Co-training, In *Proc. of the 9th Conference in Information and Knowledge and Management*, pp. 86-93.

J.-H. Oh, K.-S. Choi and H. Isahara. 2006a. A Machine Transliteration Model based on Graphemes and Phonemes, ACM TALIP, Vol. 5, No. 3, pp. 185-208.

J.-H. Oh and K.-S. Choi. 2006b. An Ensemble of Transliteration Models for Information Retrieval, In Information Processing and Management, Vol. 42, pp. 980-1002.

R. Sproat, T. Tao and C. Zhai. 2006. Named Entity Transliteration with Comparable Corpora, In *Proc. of 44th ACL,* pp. 73-80.

G. Tür, D. Hakkani-Tür and R. E. Schapire. 2005. Combining Active and Semi-supervised Learning for Spoken Language Understanding, Speech Communication, 45, pp. 171-186.

B. Vauqois. 1988. A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation, IFIP Congress-68, reprinted TAO: Vingtcinq Ans de Traduction Automatique - Analectes in C. Boitet, Ed., Association Champollin, Grenoble, pp.201-213

P. Virga and S. Khudanpur. 2003. Transliteration of Proper Names in Cross-Lingual Information Retrieval, In *Proceedings of 41st ACL Workshop on Multilingual and Mixed Language Named Entity Recognition*, pp. 57-64.

Identifying Sections in Scientific Abstracts using Conditional Random Fields

Kenji Hirohata[†]
hirohata@nii.ac.jp

Naoaki Okazaki[†]
okazaki@is.s.u-tokyo.ac.jp

Sophia Ananiadou[‡]
sophia.ananiadou@manchester.ac.uk

Mitsuru Ishizuka[†]
ishizuka@i.u-tokyo.ac.jp

[†]Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

[‡]School of Computer Science, University of Manchester
National Centre for Text Mining (NaCTeM)
Manchester Interdisciplinary Biocentre,
131 Princess Street, Manchester M1 7DN, UK

Abstract

OBJECTIVE: The prior knowledge about the rhetorical structure of scientific abstracts is useful for various text-mining tasks such as information extraction, information retrieval, and automatic summarization. This paper presents a novel approach to categorize sentences in scientific abstracts into four sections, *objective*, *methods*, *results*, and *conclusions*. **METHOD**: Formalizing the categorization task as a sequential labeling problem, we employ Conditional Random Fields (CRFs) to annotate section labels into abstract sentences. The training corpus is acquired automatically from Medline abstracts. **RESULTS**: The proposed method outperformed the previous approaches, achieving 95.5% per-sentence accuracy and 68.8% per-abstract accuracy. **CONCLUSION**: The experimental results showed that CRFs could model the rhetorical structure of abstracts more suitably.

1 Introduction

Scientific abstracts are prone to share a similar rhetorical structure. For example, an abstract usually begins with the description of background information, and is followed by the target problem, solution to the problem, evaluation of the solution, and conclusion of the paper. Previous studies observed the typical move of rhetorical roles in scientific abstracts: *problem*, *solution*, *evaluation*, and *conclusion* (Graetz, 1985; Salanger-Meyer, 1990; Swales, 1990; Orăsan, 2001). The American National Standard Institute (ANSI) recommends authors and editors of abstracts to state the *purpose*, *methods*, *results*, and *conclusions* presented in the documents (ANSI, 1979).

The prior knowledge about the rhetorical structure of abstracts is useful to improve the performance of various text-mining tasks. Marcu (1999) proposed an extraction method for summarization that captured the flow of text, based on Rhetorical Structure Theory (RST). Some extraction methods make use of cue phrases (e.g., "in conclusion", "our investigation has shown that ..."), which suggest that the rhetorical role of sentences is to identify important sentences (Edmundson, 1969; Paice, 1981). We can survey the problems, purposes, motivations, and previous approaches of a research field by reading texts in background sections of scientific papers. Tbahriti (2006) improved the performance of their information retrieval engine, giving more weight to sentences referring to *purpose* and *conclusion*.

In this paper, we present a supervised machine-learning approach that categorizes sentences in scientific abstracts into four sections, *objective*, *methods*, *results*, and *conclusions*. Figure 1 illustrates the task of this study. Given an unstructured abstract *without* section labels indicated by boldface type, the proposed method annotates section labels of each sentence. Assuming that this task is well formalized as a sequential labeling problem, we use Conditional Random Fields (CRFs) (Lafferty et al., 2001) to identify rhetorical roles in scientific abstracts.The proposed method outperforms previous approaches to this problem, achieving 95.5% per-

OBJECTIVE: This study assessed the role of adrenergic signal transmission in the control of renal erythropoietin (EPO) production in humans. **METHODS**: Forty-six healthy male volunteers underwent a hemorrhage of 750 ml. After phlebotomy, they received (intravenously for 6 hours in a parallel, randomized, placebo-controlled and single-blind design) either placebo (0.9% sodium chloride), or the beta 2-adrenergic receptor agonist fenoterol (1.5 microgram/min), or the beta 1-adrenergic receptor agonist dobutamine (5 micrograms/kg/min), or the nonselective beta-adrenergic receptor antagonist propranolol (loading dose of 0.14 mg/kg over 20 minutes, followed by 0.63 micrograms/kg/min). **RESULTS**: The AUCEPO(0-48 hr)fenoterol was 37% higher (p ¡ 0.03) than AUCEPO(0-48 hr)placebo, whereas AUCEPO(0-48 hr)dobutamine and AUCEPO(0-48 hr)propranolol were comparable with placebo. Creatinine clearance was significantly increased during dobutamine treatment. Urinary cyclic adenosine monophosphate excretion was increased only by fenoterol treatment, whereas serum potassium levels were decreased. Plasma renin activity was significantly increased during dobutamine and fenoterol infusion. **CONCLUSIONS**: This study shows in a model of controlled, physiologic stimulation of renal erythropoietin production that the beta 2-adrenergic receptor agonist fenoterol but not the beta 1-adrenergic receptor agonist dobutamine is able to increase erythropoietin levels in humans. The result can be interpreted as a hint that signals for the control of erythropoietin production may be mediated by beta 2-adrenergic receptors rather than by beta 1-adrenergic receptors. It appears to be unlikely that an increase of renin concentrations or glomerular filtration rate is causally linked to the control of erythropoietin production in this experimental setting.

Figure 1: An abstract with section labels indicated by boldface type (Gleiter et al., 1997).

sentence accuracy and 68.8% per-abstract accuracy.

This paper is organized as follows. Section 2 describes previous approaches to this task. Formalizing the task as a sequential-labeling problem, Section 3 designs a sentence classifier using CRFs. Training corpora for the classifier are acquired automatically from the Medline abstracts. Section 4 reports considerable improvements in the proposed method over the baseline method using Support Vector Machine (SVM) (Cortes and Vapnik, 1995). We conclude this paper in Section 5.

2 Related Work

The previous studies regarded the task of identifying section names as a text-classification problem that determines a label (section name) for each sentence. Various classifiers for text categorization, Naïve Bayesian Model (NBM) (Teufel and Moens, 2002; Ruch et al., 2007), Hidden Markov Model (HMM) (Wu et al., 2006; Lin et al., 2006), and Support Vector Machines (SVM) (McKnight and Arinivasan, 2003; Shimbo et al., 2003; Ito et al., 2004; Yamamoto and Takagi, 2005) were applied.

Table 1 summarizes these approaches and performances. All studies target scientific abstracts except for Teufel and Moens (2002) who target scientific full papers. Field *classes* show the set of section names that each study assumes: background (B), objective/aim/purpose (O), method (M), result (R), conclusion (C), and introduction (I) that combines the background and objective. Although we should not compare directly the performances of these studies, which use a different set of classification labels and evaluation corpora, SVM classifiers appear to yield better results for this task. The rest of this section elaborates on the previous studies with SVMs.

Shimbo et al. (2003) presented an advanced text retrieval system for Medline that can focus on a specific section in abstracts specified by a user. The system classifies sentences in each Medline abstract into four sections, *objective*, *method*, *results*, and *conclusion*. Each sentence is represented by words, word bigrams, and contextual information of the sentence (e.g., class of the previous sentence, relative location of the current sentence). They reported 91.9% accuracy (per-sentence basis) and 51.2% accuracy (per-abstract basis[1]) for the classification with the best feature set for quadratic SVM. Ito et al. (2004) extended the work with a semi-supervised learning technique using transductive SVM (TSVM).

Yamamoto and Takagi (2005) developed a system to classify abstract sentences into five sections, *background*, *purpose*, *method*, *result*, and *conclusion*. They trained a linear-SVM classifier with features such as unigram, subject-verb, verb tense, relative sentence location, and sentence score (average TF*IDF score of constituent words). Their method achieved 68.9%, 63.0%, 83.6%, 87.2%, 89.8% F-scores for classifying *background*, *purpose*, *method*, *result*, and *conclusion* sentences respectively. They also reported the classification performance of *introduction* sentences, which combines *background* and *purpose* sentences, with 91.3% F-score.

[1] An abstract is considered correct if all constituent sentences are correctly labeled.

Methods	Model	Classes	Performance (reported in papers)
Teufel and Moens (2002)	NBM	(7 classes)	44% precision and 65% recall for *aim* sentences
Ruch et al. (2007)	NBM	O M R C	85% F-score for *conclusion* sentences
Wu et al. (2006)	HMM	B O M R C	80.54% precision
Lin et al. (2006)	HMM	I M R C	88.5%, 84.3%, 89.8%, 89.7% F-scores
McKnight and Srinivasan (2003)	SVM	I M R C	89.2%, 82.0%, 82.1%, 89.5% F-scores
Shimbo et al. (2003)	SVM	B O M R C	91.9% accuracy
Ito et al. (2004)	TSVM	B O M R C	66.0%, 51.0%, 49.3%, 72.9%, 67.7% F-scores
Yamamoto and Takagi (2005)	SVM	I (B O) M R C	91.3% (68.9%, 63.0%), 83.6%, 87.2%, 89.8% F-scores

Table 1: Approaches and performances of previous studies on section identification

3 Proposed method

3.1 Section identification as a sequence labeling problem

The previous work saw the task of labeling as a text categorization that determines the class label y_i for each sentence x_i. Even though some work includes features of the surrounding sentences for x_i, e.g. "class label of x_{i-1} sentence," "class label of x_{i+1} sentence," and "unigram in x_{i-1} sentence," the classifier determines the class label y_i for each sentence x_i independently. It has been an assumption for text classification tasks to decide a class label independently of other class labels.

However, as described in Section 1, scientific abstracts have typical moves of rhetorical roles: it would be very peculiar if *result* sentences appearing before *method* sentences were described in an abstract. Moreover, we would like to model the structure of abstract sentences rather than modeling just the section label for each sentence. Thus, the task is more suitably formalized as a sequence labeling problem: given an abstract with sentences $\boldsymbol{x} = (x_1, ..., x_n)$, determine the optimal sequence of section names $\boldsymbol{y} = (y_1, ..., y_n)$ of all possible sequences.

Conditional Random Fields (CRFs) have been successfully applied to various NLP tasks including part-of-speech tagging (Lafferty et al., 2001) and shallow parsing (Sha and Pereira, 2003). CRFs define a conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ for output and input sequences, $\boldsymbol{y}$ and $\boldsymbol{x}$,

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z_{\boldsymbol{\lambda}}(\boldsymbol{x})} \exp\left\{\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})\right\}. \quad (1)$$

Therein: function $\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})$ denotes a global feature vector for input sequence $\boldsymbol{x}$ and output sequence $\boldsymbol{y}$,

$$\boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x}) = \sum_i \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x}, i), \quad (2)$$

i ranges over the input sequence, function $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x}, i)$ is a feature vector for input sequence $\boldsymbol{x}$ and output sequence $\boldsymbol{y}$ at position i (based on state features and transition features), $\boldsymbol{\lambda}$ is a vector where an element $\boldsymbol{\lambda}_k$ represents the weight of feature $\boldsymbol{F}_k(\boldsymbol{y}, \boldsymbol{x})$, and $Z_{\boldsymbol{\lambda}}(\boldsymbol{x})$ is a normalization factor,

$$Z_{\boldsymbol{\lambda}}(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp\left\{\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{x})\right\}. \quad (3)$$

The optimal output sequence $\hat{\boldsymbol{y}}$ for an input sequence $\boldsymbol{x}$,

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}), \quad (4)$$

is obtained efficiently by the Viterbi algorithm. The optimal set of parameters $\boldsymbol{\lambda}$ is determined efficiently by the Generalized Iterative Scaling (GIS) (Darroch and Ratcliff, 1972) or Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) (Nocedal and Wright, 1999) method.

3.2 Features

We design three kinds of features to represent each abstract sentence for CRFs. The contributions of these features will be evaluated later in Section 4.

Content (n-gram) This feature examines the existence of expressions that characterize a specific section, e.g. "to determine ...," and "aim at ..." for stating the *objective* of a study. We use features for sentence contents represented by: i) words, ii) word bigrams, and iii) mixture of words and word bigrams. Words are normalized into their base forms by the GENIA tagger (Tsuruoka and Tsujii, 2005), which is a part-of-speech tagger trained for the biomedical

Rank	OBJECTIVE	METHOD	RESULTS	CONCLUSIONS
1	# to	be measure	%)	suggest that
2	be to	be perform	(p	may be
3	to determine	n =	p <	# these
4	study be	be compare	) .	should be
5	this study	be determine	% .	these result

Table 2: Bigram features with high χ^2 values ('#' stands for a beginning of a sentence).

domain. We measure the co-occurrence strength (χ^2 value) between each feature and section label. If a feature appears selectively in a specific section, the χ^2 value is expected to be high. Thus, we extract the top 200,000 features[2] that have high χ^2 values to reduce the total number of features. Table 3.2 shows examples of the top five bigrams that have high χ^2 values.

Relative sentence location An abstract is likely to state *objective* of the study at the beginning and its *conclusion* at the end. The position of a sentence may be a good clue for determining its section label. Thus, we design five binary features to indicate relative position of sentences in five scales.

Features from previous/next w sentences This reproduces features from previous and following w sentences to the current sentence ($w = \{0, 1, 2\}$), so that a classifier can make use of the content of the surrounding sentences. Duplicated features have prefixes (e.g. `PREV_` and `NEXT_`) to distinguish their origins.

3.3 Section labels

It would require much effort and time to prepare a large amount of abstracts annotated with section labels. Fortunately, some Medline abstracts have section labels stated explicitly by its authors. We examined section labels in 7,811,582 abstracts in the whole Medline[3], using the regular-expression pattern:

```
^[A-Z]+([ ][A-Z]+){0,3}:[ ]
```

A sentence is qualified to have a section name if it begins with up to 4 uppercase token(s) followed by a colon ':'. This pattern identified 683,207 (ca. 9%) abstracts with structured sections.

Table 3 shows typical moves of sections in Medline abstracts. The majority of sequences in this table consists of four sections compatible with the ANSI standard, *purpose*, *methods*, *results*, and *conclusions*. Moreover, the most frequent sequence is "OBJECTIVE → METHOD(S) → RESULTS → CONCLUSION(S)," supposing that AIM and PURPOSE are equivalent to OBJECTIVE. Hence, this study assumes four sections, OBJECTIVE, METHOD, RESULTS, and CONCLUSIONS.

Meanwhile, it is common for NP chunking tasks to represent a chunk (e.g., NP) with two labels, the *begin* (e.g., B-NP) and *inside* (e.g., I-NP) of a chunk (Ramshaw and Marcus, 1995). Although none of the previous studies employed this representation, attaching B- and I- prefixes to section labels may improve a classifier by associating clue phrases (e.g., "to determine") with the starts of sections (e.g., B-OBJECTIVE). We will compare classification performances on two sets of label representations: namely, we will compare four section labels and eight labels with BI prefixes attached to section names.

4 Evaluation

4.1 Experiment

We constructed two sets of corpora ('pure' and 'expanded'), each of which contains 51,000 abstracts sampled from the abstracts with structured sections. The 'pure' corpus consists of abstracts that have the exact four section labels. In other words, this corpus does not include AIM or PURPOSE sentences even though they are equivalent to OBJECTIVE sentences. The 'pure' corpus is useful to compare the performance of this study with the previous work.

[2]We chose the number of features based on exploratory experiments.

[3]The Medline database was up-to-date on March 2006.

Rank	# abstracts	(%)	Section sequence
1	111,617	(17.6)	OBJECTIVE → METHOD(S) → RESULT(S) → CONCLUSION(S)
2	107,124	(16.9)	BACKGROUND(S) → METHOD(S) → RESULT(S) → CONCLUSION(S)
3	40,083	(6.3)	PURPOSE → METHOD(S) → RESULT(S) → CONCLUSION(S)
4	20,519	(3.2)	PURPOSE → MATERIAL AND METHOD(S) → RESULT(S) → CONCLUSION(S)
5	16,705	(2.6)	AIM(S) → METHOD(S) → RESULT(S) → CONCLUSION(S)
6	16,400	(2.6)	BACKGROUND → OBJECTIVE → METHOD(S) → RESULT(S) → CONCLUSION(S)
7	12,227	(1.9)	OBJECTIVE → STUDY DESIGN → RESULT(S) → CONCLUSION(S)
8	11,483	(1.8)	BACKGROUND →METHOD(S) AND RESULT(S) → CONCLUSION(S)
9	8,866	(1.4)	OBJECTIVE → MATERIAL AND METHOD(S) → RESULT(S) → CONCLUSION(S)
10	8,537	(1.3)	PURPOSE → PATIENT AND METHOD(S) → RESULT(S) → CONCLUSION(S)
..	...	...	...
Total	683,207	(100.0)	

Table 3: Typical sequences of sections in Medline abstracts

Representative	Equivalent section labels
OBJECTIVE	AIM, AIM OF THE STUDY, AIMS, BACKGROUND/AIMS, BACKGROUND/PURPOSE, BACKGROUND, BACKGROUND AND AIMS, BACKGROUND AND OBJECTIVE, BACKGROUND AND OBJECTIVES, BACKGROUND AND PURPOSE, CONTEXT, INTRODUCTION, OBJECT, OBJECTIVE, OBJECTIVES, PROBLEM, PURPOSE, STUDY OBJECTIVE, STUDY OBJECTIVES, SUMMARY OF BACKGROUND DATA
METHOD	ANIMALS, DESIGN, DESIGN AND METHODS, DESIGN AND SETTING, EXPERIMENTAL DESIGN,INTERVENTION, INTERVENTION(S), INTERVENTIONS, MATERIAL AND METHODS, MATERIALS AND METHODS, MEASUREMENTS, METHOD, METHODOLOGY, METHODS, METHODS AND MATERIALS, PARTICIPANTS, PATIENT(S), PATIENTS, PATIENTS AND METHODS, PROCEDURE, RESEARCH DESIGN AND METHODS, SETTING, STUDY DESIGN, STUDY DESIGN AND METHODS, SUBJECTS, SUBJECTS AND METHODS
RESULTS	FINDINGS, MAIN RESULTS, RESULT, RESULT(S), RESULTS
CONCLUSIONS	CONCLUSION, CONCLUSION(S), CONCLUSIONS, CONCLUSIONS AND CLINICAL RELEVANCE, DISCUSSION, IMPLICATIONS, INTERPRETATION, INTERPRETATION AND CONCLUSIONS

Table 4: Representative section names and their expanded sections

In contrast, the 'expanded' corpus includes sentences in equivalent sections: AIM and PURPOSE sentences are mapped to the OBJECTIVE. Table 4 shows the sets of equivalent sections for representative sections. We created this mapping table manually by analyzing the top 100 frequent section labels found in the Medline. The 'expanded' corpus is close to the real situation in which the proposed method annotates unstructured abstracts.

We utilized FlexCRFs[4] implementation to build a classifier with linear-chain CRFs. As a baseline method, we also prepared an SVM classifier[5] with the same features.

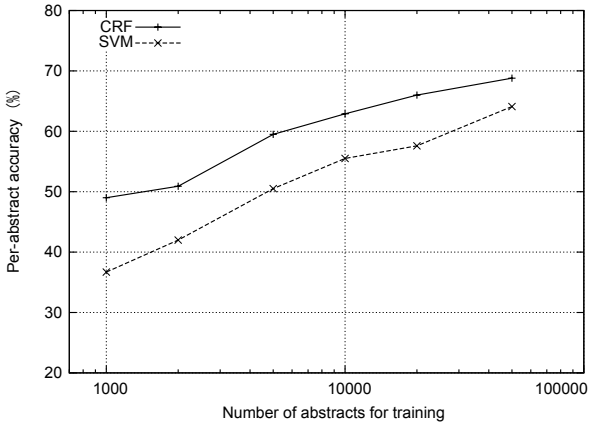


Figure 2: Training curve

4.2 Results

Given the number of abstracts for training n, we randomly sampled n abstracts from a corpus for training and 1,000 abtracts for testing. Content (n-gram) features were generated for each trainig set. We

[4]Flexible Conditional Random Field Toolkit (FlexCRFs): `http://flexcrfs.sourceforge.net/`

[5]We used SVM^{light} implementation with the linear kernel, which achieved the best accuracy through this experiment: `http://svmlight.joachims.org/`

Section labels	With B- and I- prefixes		Without B- and I- prefixes	
Features	CRF	SVM	CRF	SVM
n-gram	88.7 (42.4)	81.5 (19.1)	85.7 (33.0)	83.3 (23.4)
n-gram + position	93.4 (59.7)	88.2 (35.5)	92.4 (55.4)	89.6 (39.4)
n-gram + surrounding ($w = 1$)	93.3 (60.4)	89.9 (42.2)	92.1 (52.8)	90.0 (42.0)
n-gram + surrounding ($w = 2$)	93.7 (61.1)	91.8 (49.4)	92.8 (54.3)	91.8 (47.0)
Full	94.3 (62.9)	93.3 (55.5)	93.3 (56.1)	92.9 (52.2)

Table 5: Classification performance (accuracy) on 'pure' corpus ($n = 10,000$)

Section labels	With B- and I- prefixes		Without B- and I- prefixes	
Features	CRF	SVM	CRF	SVM
n-gram	87.7 (35.6)	78.5 (14.5)	81.9 (21.0)	80.0 (16.2)
n-gram + position	92.6 (54.3)	87.1 (31.2)	91.4 (48.7)	88.1 (31.2)
n-gram + surrounding ($w = 1$)	92.3 (52.0)	88.5 (37.6)	89.9 (44.0)	88.4 (37.1)
n-gram + surrounding ($w = 2$)	92.4 (52.5)	90.1 (41.1)	91.2 (46.6)	90.4 (41.6)
Full	93.0 (55.0)	92.0 (47.3)	92.5 (50.9)	91.7 (44.0)

Table 6: Classification performance (accuracy) on 'expanded' corpus ($n = 10,000$)

measured the classification accuracy of sentences (per-sentence accuracy) and abstracts (per-abstract accuracy). In per-abstract accuracy, an abstract is considered correct if all constituent sentences are correctly labeled.

Trained with $n = 50,000$ abstracts from 'pure' corpus, the proposed method achieved 95.5% per-sentence accuracy and 68.8% per-abstract accuracy. The F-score for each section label was 98.7% (O), 95.8% (M), 95.0% (R), and 94.2% (C). The proposed method performed this task better than the previous studies by a great margin. Figure 2 shows the training curve for the 'pure' corpus with all features presented in this paper. CRF and SVM methods performed better with more abstracts used for training. This training curve demonstrated that, with less than half the number of training corpus, the proposed method could achieve the same accuracy as the baseline method.

Tables 5 and 6 report the performance of the proposed and baseline methods on 'pure' and 'expanded' corpora respectively ($n = 10,000$). These tables show per-sentence accuracy followed by per-abstract accuracy in parentheses with different configurations of features (row) and label representations (column). For example, the proposed method obtained 94.3% per-sentence accuracy and 62.9% per-abstract accuracy with 10,000 training abstracts from 'pure' corpus, all features, and BI prefixes for class labels.

The proposed method outperformed the baseline method in all experimental configurations. This suggests that CRFs are more suitable for modeling moves of rhetorical roles in scientific abstracts. It is noteworthy that the CRF classifier gained higher per-abstract accuracy than the SVM. For example, both the CRF classifier with features from surrounding sentences ($w = 1$), and SVM classifier with full features, obtained 93.3% per-sentence accuracy in Table 5. Nevertheless, the per-abstract accuracies of the former and latter were 60.4% and 55.5% respectively: the CRF classifier had roughly 5% advantage on per-abstract accuracy over SVM. This analysis reflects the capability of CRFs to determine the optimal sequence of section names.

Additional features such as sentence position and surrounding sentences improved the performance by ca. 5–10%. The proposed method achieved the best results with all features. Another interesting discussion arises with regard to the representations of section labels. The BI representation always boosted the per-abstract accuracy of CRF classifiers by ca. 4–14%. In contrast, the SVM classifier could not leverage the BI representation, and in some configurations, even degraded the accuracy.

5 Conclusion

This paper presented a novel approach to identifying rhetorical roles in scientific abstracts using CRFs. The proposed method achieved more successful results than any other previous reports. The CRF classifier had roughly 5% advantage on per-abstract accuracy over SVM. The BI representation of section names also boosted the classification accuracy by 5%. In total, the proposed method gained more than 10% improvement on per-abstract accuracy.

We have evaluated the proposed method only on medical literatures. In addition to improving the classification performance, a future direction for this study would be to examine the adaptability of the proposed method to include other types of texts. We are planning to construct a summarization system using the proposed method.

References

ANSI. 1979. American national standard for writing abstracts. Z39.14-1979, American National Standards Institute (ANSI).

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

John N. Darroch and Douglas Ratcliff. 1972. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

Harold P. Edmundson. 1969. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Christoph H. Gleiter, Tilmann Becker, Katharina H. Schreeb, Stefan Freudenthaler, and Ursula Gundert-Remy. 1997. Fenoterol but not dobutamine increases erythropoietin production in humans. *Clinical Pharmacology & Therapeutics*, 61(6):669–676.

Naomi Graetz. 1985. Teaching EFL students to extract structural information from abstracts. In Jan M. Ulijn and Anthony K. Pugh, editors, *Reading for Professional Purposed: Methods and Materials in Teaching Languages*, pages 123–135. Acco, Leuven, Belgium.

Takahiko Ito, Masashi Simbo, Takahiro Yamasaki, and Yuji Matsumoto. 2004. Semi-supervised sentence classification for medline documents. In *IPSJ SIG Technical Report*, volume 2004-ICS-138, pages 141–146.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML-2001)*, pages 282–289.

Jimmy Lin, Damianos Karakos, Dina Demner-Fushman, and Sanjeev Khudanpur. 2006. Generative content models for structural analysis of medical abstracts. In *Proceedings of the HLT/NAACL 2006 Workshop on Biomedical Natural Language Processing (BioNLP'06)*, pages 65–72, New York City, USA.

Daniel Marcu. 1999. Discourse trees are good indicators of importance in text. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automatic Text Summarization*. MIT Press.

Larry McKnight and Padmini Arinivasan. 2003. Categorization of sentence types in medical abstracts. In *AMIA 2003 Symposium Proceedings*, pages 440–444.

Jorge Nocedal and Stephen J. Wright. 1999. *Numerical Optimization*. Springer-Verlag, New York, USA.

Constantin Orăsan. 2001. Patterns in scientific abstracts. In *Proceedings of Corpus Linguistics 2001 Conference*, pages 433 – 443, Lancaster University, Lancaster, UK.

Chris D. Paice. 1981. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 172–191, Kent, UK. Butterworth & Co.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the ACL 3rd Workshop on Very Large Corpora*, pages 82–94.

Patrick Ruch, Celia Boyer, Christine Chichester, Imad Tbahriti, Antoine Geissbühler, Paul Fabry, Julien Gobeill, Violaine Pillet, Dietrich Rebholz-Schuhmann, Christian Lovis, and Anne-Lise Veuthey. 2007. Using argumentation to extract key sentences from biomedical abstracts. *International Journal of Medical Informatics*, 76(2–3):195–200.

Françoise Salanger-Meyer. 1990. Discoursal flaws in medical english abstracts: A genre analysis per research- and text-type. *Text*, 10(4):365–384.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141, Edmonton, Canada.

Masashi Shimbo, Takahiro Yamasaki, and Yuji Matsumoto. 2003. Using sectioning information for text retrieval: a case study with the medline abstracts. In *Proceedings of Second International Workshop on Active Mining (AM'03)*, pages 32–41.

John M. Swales, 1990. *Genre Analysis: English in academic and research settings*, chapter 6. Cambridge University Press, UK.

Imad Tbahriti, Christine Chichester, Frédérique Lisacek, and Patrick Ruch. 2006. Using argumentation to retrieve articles with similar citations: An inquiry into improving related articles search in the medline digital library. *International Journal OF Medical Informatics*, 75(6):488–495.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 467–474, Vancouver, British Columbia, Canada.

Jien-Chen Wu, Yu-Chia Chang, Hsien-Chin Liou, and Jason S. Chang. 2006. Computational analysis of move structures in academic abstracts. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 41–44, Sydney, Australia.

Yasunori Yamamoto and Toshihisa Takagi. 2005. A sentence classification system for multi-document summarization in the biomedical domain. In *Proceedings of the International Workshop on Biomedical Data Engineering (BMDE2005)*, pages 90–95.

Formalising Multi-layer Corpora in OWL DL – Lexicon Modelling, Querying and Consistency Control

Aljoscha Burchardt[1], **Sebastian Padó**[2*], **Dennis Spohr**[3*], **Anette Frank**[4*] **and Ulrich Heid**[3]

[1]Dept. of Comp. Ling.
Saarland University
Saarbrücken, Germany
albu@coli.uni-sb.de

[2]Dept. of Linguistics
Stanford University
Stanford, CA
pado@stanford.edu

[3]Inst. for NLP
University of Stuttgart
Stuttgart, Germany
spohrds,heid@ims.uni-stuttgart.de

[4]Dept. of Comp. Ling.
University of Heidelberg
Heidelberg, Germany
frank@cl.uni-heidelberg.de

Abstract

We present a general approach to formally modelling corpora with multi-layered annotation, thereby inducing a *lexicon model* in a typed logical representation language, OWL DL. This model can be interpreted as a graph structure that offers *flexible querying functionality* beyond current XML-based query languages and powerful methods for *consistency control*. We illustrate our approach by applying it to the syntactically and semantically annotated SALSA/TIGER corpus.

1 Introduction

Over the years, much effort has gone into the creation of large corpora *with multiple layers of linguistic annotation*, such as morphology, syntax, semantics, and discourse structure. Such corpora offer the possibility to empirically investigate the interactions between different levels of linguistic analysis.

Currently, the most common use of such corpora is the acquisition of statistical models that make use of the "more shallow" levels to predict the "deeper" levels of annotation (Gildea and Jurafsky, 2002; Miltsakaki et al., 2005). While these models fill an important need for practical applications, they fall short of the general task of *lexicon modelling*, i.e., creating an abstracted and compact representation of the corpus information that lends itself to 'linguistically informed' usages such as human interpretation or integration with other knowledge sources (e.g., deep grammar resources or ontologies). In practice, this task faces three major problems:

Ensuring consistency. Annotation reliability and consistency are key prerequisites for the extraction of generalised linguistic knowledge. However, with the increasing complexity of annotations for 'deeper' (in particular, semantic) linguistic analysis, it becomes more difficult to ensure that all annotation instances are consistent with the annotation scheme.

Querying multiple layers of linguistic annotation. A recent survey (Lai and Bird, 2004) found that currently available XML-based corpus query tools support queries operating on multiple linguistic levels only in very restricted ways. Particularly problematic are intersecting hierarchies, i.e., tree-shaped analyses on multiple linguistic levels.

Abstractions and application interfaces. A pervasive problem in annotation is granularity: The granularity offered by a given annotation layer may diverge considerably from the granularity that is needed for the integration of corpus-derived data in large symbolic processing architectures or general lexical resources. This problem is multiplied when more than one layer of annotation is considered, for example in the characterisation of interface phenomena. While it may be possible to obtain coarser-grained representations procedurally by collapsing categories, such procedures are not flexibly configurable.

Figure 1 illustrates these difficulties with a sentence from the SALSA/TIGER corpus (Burchardt et al., 2006), a manually annotated German newspaper corpus which contains role-semantic analyses in the FrameNet paradigm (Fillmore et al., 2003) on top of syntactic structure (Brants et al., 2002).[1] The se-

[*]At the time of writing, Sebastian Padó and Dennis Spohr were affiliated with Saarland University, and Anette Frank with DFKI Saarbrücken and Saarland University.

[1]While FrameNet was originally developed for English, the majority of frames has been found to generalise well to other

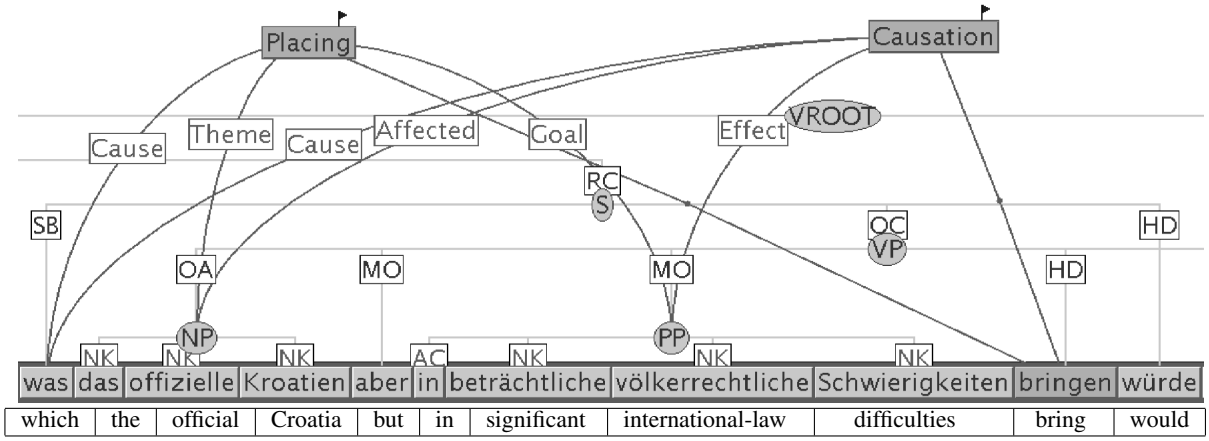


Figure 1: Multi-layer annotation of a German phrase with syntax and frame semantics *('which would bring official Croatia into significant difficulties with international law')*

mantic structure consists of *frames*, semantic classes assigned to predicating expressions, and the semantic roles introduced by these classes. The verb *bringen ('to bring')* is used metaphorically and is thus analysed as introducing one frame for the "literal" reading (PLACING) and one for the "understood" reading (CAUSATION), both with their own role sets.

The high complexity of the semantic structure even on its own shows the necessity of a device for consistency checking. In conjunction with syntax, it presents exactly the case of intersecting hierarchies which is difficult to query. With respect to the issue of abstraction, note that semantic roles are realised variously as individual words (*was ('which')*) and constituents (NPs, PPs), a well-known problem in deriving syntax-semantics mappings from corpora (Frank, 2004; Babko-Malaya et al., 2006).

Our proposal. We propose that the problems introduced above can be addressed by formalising corpora in an *integrated, multi-layered corpus and lexicon model* in a declarative logical framework, more specifically, the description logics-based OWL DL formalism. The major benefits of this approach are that all relevant properties of the annotation *and* the underlying model are captured in a uniform representation and, moreover, that the formal semantics of the model makes it possible to use general and efficient knowledge representation techniques for consistency control. Finally, we can extract specific *subsets* from a corpus by defining *task-specific views* on the graph.

After a short discussion of related approaches in Section 2, Section 3 provides details on our methodology. Sections 4 and 5 demonstrate the benefits of our strategy on a model of the SALSA/TIGER data. Section 6 concludes.

2 Related Work

One recent approach to lexical resource modelling is the Lexical Systems framework (Polguère, 2006), which aims at providing a highly general representation for arbitrary kinds of lexica. While this is desirable from a representational point of view, the resulting models are arguably too generic to support strong consistency checks on the encoded data.

A further proposal is the currently evolving Lexical Markup Framework (LMF; Francopoulo et al. (2006)), an ISO standard for lexical resource modelling, and an LMF version of FrameNet exists. However, we believe that our usage of a typed formalism takes advantage of a strong logical foundation and the notions of inheritance and entailment (cf. Scheffczyk et al. (2006)) and is a crucial step beyond the representational means provided by LMF.

Finally, the closest neighbour to our proposal is the ATLAS project (Laprun et al., 2002), which combines annotations with a descriptive meta-model. However, to our knowledge, ATLAS only models basic consistency constraints, and does not capture dependencies between different layers of annotation.

languages (Burchardt et al., 2006; Boas, 2005).

3 Modelling Multilevel Corpora in OWL DL

3.1 A formal graph-based Lexicon

This section demonstrates how OWL DL, a strongly typed representation language, can serve to transparently formalise corpora with multi-level annotation. OWL DL is a logical language that combines the expressivity of OWL[2] with the favourable computational properties of Description Logics (DL), notably decidability and monotonicity (Baader et al., 2003). The strongly typed, well-defined model-theoretic semantics distinguishes OWL DL from recent alternative approaches to lexicon modelling.

Due to the fact that OWL DL has been defined in the Resource Description Framework (RDF[3]), the first central benefit of using OWL DL is the possibility to conceive of the lexicon as a *graph* – a net-like entity with a high degree of interaction between layers of linguistic description, with an associated class hierarchy. Although OWL DL itself does not have a graph model but a model-theoretic semantics based on First Order Logic, we will illustrate our ideas with reference to a graph-like representation, since this is what we obtain by transforming our OWL DL files into an RDFS database.

Each node in the graph instantiates one or more classes that determine the *properties* of the node. In a straightforward sense, properties correspond to labelled edges between nodes. They are, however, also represented as nodes in the graph which instantiate (meta-)classes themselves.

The model is kept compact by OWL's support for *multiple instantiation*, i.e., the ability of instances to realise more than one class. For example, in a syntactically and semantically annotated corpus, all syntactic units (constituents, words, or even parts of words) can instantiate – in addition to a syntactic class – one or more semantic classes. Multiple instantiation enables the representation of information about several annotation layers within single instances.

As we have argued in Section 2, we believe that having one generic model that can represent all corpora is problematic. Instead, we propose to construct lexicon models for specific types of corpora. The design of such models faces two central design questions: (a) Which properties of the annotated instances should be represented?; (b) How are different types of these annotation properties modelled in the graph?

Implicit features in annotations. Linguistic annotation guidelines often concentrate on specifying the *linguistic data categories* to be annotated. However, a lot of linguistically relevant information often remains implicit in the annotation scheme. Examples from the SALSA corpus include, e.g., the fact that the annotation in Figure 1 is metaphorical. This information has to be inferred from the configuration that one predicate evokes two frames. As such information about different annotation types is useful in final lexicon resources, e.g. to define clean generalisations over the data (singling out "special cases"), to extract information about special data categories, and to define formally grounded consistency constraints, we include it in the lexicon model.

Form of representation. All relevant information has to be represented either as assertional statements in the model graph (i.e., nodes connected by edges), or as definitional axioms in the class hierarchy.[4]

This decision involves a fundamental trade-off between expressivity and flexibility. Modelling features as axioms in the class hierarchy imposes definitional constraints on all instances of these classes and is arguably more attractive from a cognitive perspective. However, modelling features as entities in the graph leads to a smaller class hierarchy, increased querying flexibility, and more robustness in the face of variation and noise in the data.

3.2 Modelling SALSA/TIGER Data

We now illustrate these decisions concretely by designing a model for a corpus with syntactic and frame-semantic annotation, more concretely the SALSA/TIGER corpus. However, the general points we make are valid beyond this particular setting.

As concerns implicit annotation features, we have designed a *hierarchy of annotation types* which now explicitly expresses different classes of annotation phenomena and which allows for the definition of annotation class-specific properties. For example, frame targets are marked as a multi-word target if

[2] http://www.w3.org/2004/OWL/

[3] http://www.w3.org/RDF/

[4] This choice corresponds to the DL distinction between TBox ("intensional knowledge") and ABox ("extensional knowledge").

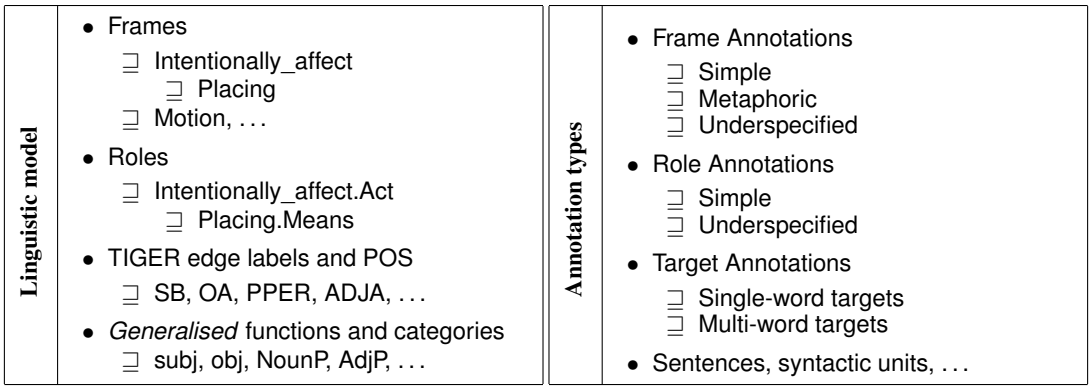


Figure 2: Schema of the OWL DL model's class hierarchy ("TBox")

their span contains at least two terminal nodes. The hierarchy is shown on the right of Figure 2, which shows parts of the bipartite class hierarchy.

The left-hand side of Figure 2 illustrates the *linguistic model*, in which frames and roles are organised according to FrameNet's inheritance relation. Although this design seems to be straightforward, it is the result of careful considerations concerning the second design decision. Since FrameNet is a hierarchically structured resource with built-in inheritance relations, one important question is whether to model individual frames, such as SELF_MOTION or LEADERSHIP, and their relations either as instances of a general class `Frame` and as links between these instances, or as hierarchically structured classes with richer axiomatisation. In line with our focus on consistency checking, we adopt the latter option, which allows us to use built-in reasoning mechanisms of OWL DL to ensure consistency.

Annotation instances from the corpus instantiate multiple classes in both hierarchies (cf. Figure 2): On the annotation side according to their types of phenomena; on the linguistic side based on their frames, roles, syntactic functions, and categories.

Flexible abstraction. Section 1 introduced granularity as a pervasive problem in the use of multi-level corpora. Figure 2 indicates that the class hierarchy of the OWL DL model offers a very elegant way of defining *generalised* data categories that provide abstractions over model classes, both for linguistic categories and annotation types. Moreover, properties can be added to each abstracting class and then be used, e.g., for consistency checking. In our case, Figure 2 shows (functional) edge labels and part-of-speech tags provided by TIGER, as well as sets of (largely theory-neutral) grammatical functions and categories that subsume these fine-grained categories and support the extraction of generalised valence information from the lexicon.

An annotated corpus sentence. To substantiate the above discussion, Figure 3 shows a partial lexicon representation of the example in Figure 1. The boxes represent instance nodes, with classes listed above the horizontal line, and datatype properties below it.[5] The links between these instances indicate OWL object properties which have been defined for the instantiated classes. For example, the metaphorical PLACING frame is shown as a grey box in the middle.

Multiple inheritance is indicated by instances carrying more than one class, such as the instance in the left centre, which instantiates the classes `SyntacticUnit`, `NP`, `OA`, `NounP` and `obj`. Multi-class instances inherit the properties of each of these classes, so that e.g., the metaphoric frame annotation of the PLACING frame in the middle has both the properties defined for *frames* (`hasCoreRole`) and for *frame annotations* (`hasTarget`). The generalised syntactic categories discussed above are given in italics (e.g., *NounP*).

The figure highlights the model's graph-based structure with a high degree of interrelation between the lexicon entities. For example, the grey PLACING frame instance is directly related to its roles (left, bottom), its lexical anchor (right), the surrounding sentence (top), and a flag (top left) indicating metaphorical use.

[5] For the sake of simplicity, we excluded explicit 'is-a' links.

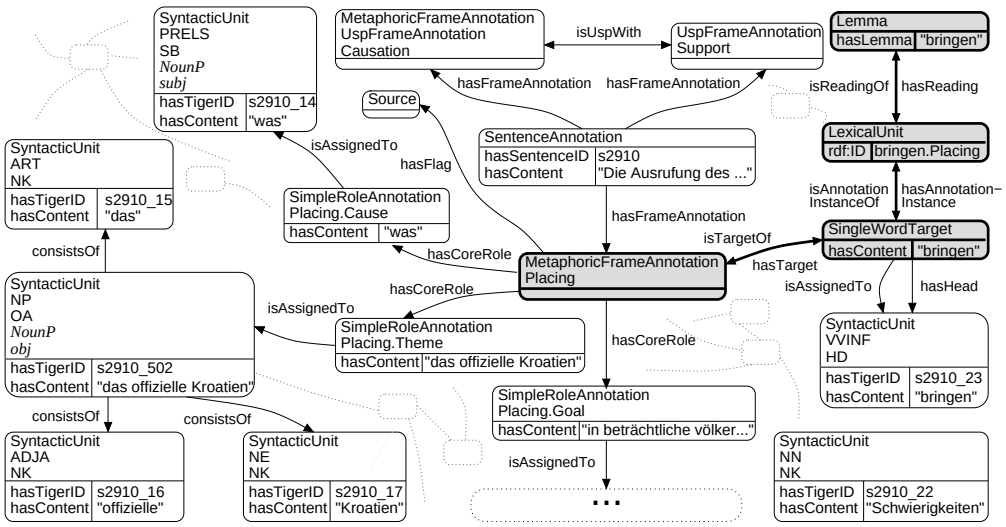


Figure 3: Partial lexicon representation of an annotated corpus sentence

4 Querying the Model

We now address the second desideratum introduced in Section 1, namely a flexible and powerful query mechanism. For OWL DL models, such a mechanism is available in the form of the Sesame (Broekstra et al., 2002) SeRQL query language. Since SeRQL makes it possible to extract and view arbitrary subgraphs of the model, querying of intersective hierarchies is possible in an intuitive manner.

An interesting application for this querying mechanism is to extract genuine *lexicon views* on the corpus annotations, e.g., to extract syntax-semantics mapping information for particular senses of lemmas, by correlating role assignments with deep syntactic information. These can serve both for inspection and for interfacing the annotation data with deep grammatical resources or general lexica. Applied to our complete corpus, this "lexicon" contains on average 8.5 role sets per lemma, and 5.6 role sets per frame. The result of such a query is illustrated in Table 1 for the lemma *senken ('to lower').*

From such view, frame- or lemma-specific *role sets*, i.e., patterns of role-category-function assignments can easily be retrieved. A typical example is given in Table 2, with additional frequency counts. The first row indicates that the AGENT role has been realised as a (deep) subject noun phrase and the ITEM as (deep) object noun phrase.

We found that generalisations over corpus categories encoded in the class hierarchies are central

Role	*Cat*	*Func*	*Freq*
Item	NounP	obj	26
Agent	NounP	subj	15
Difference	PrepP	mod-um	6
Cause	NounP	subj	4
Value_2	PrepP	mod-auf	3
Value_2	PrepP	pobj-auf	2
Value_1	PrepP	mod-von	1

Table 1: Role-category-function assignments for *senken* / CAUSE_CHANGE_OF_SCALAR_POSITION (CCSP)

Role set for senken / CCSP			*Freq*
Agent subj NounP		Item obj NounP	11
	Cause subj NounP	Item obj NounP	4
		Item obj NounP	4
Agent subj NounP		Item obj NounP	Difference mod-um PrepP	2

Table 2: Sample of role sets for *senken* / CCSP

to the usefulness of the resulting patterns. For example, the number of unique mappings between semantic roles and syntactic categories in our corpus is 5,065 for specific corpus categories, and 2,289 for abstracted categories. Thus, the definition of an abstraction layer, in conjunction with a flexible query mechanism, allows us to induce lexical characterisations of the syntax-semantics mapping – aggregated

and generalised from disparate corpus annotations.

Incremental refinements. Querying, and the resulting lexical views, can serve yet another purpose: Such aggregates make it possible to conduct a *data-driven* search for linguistic generalisations which might not be obvious from a theoretical perspective, and allow quick inspection of the data for counterexamples to plausible regularities.

In the case of semantic roles, for example, such a regularity would be that semantic roles are not assigned to conflicting grammatical functions (e.g., deep subject and object) within a given lemma. However, some of the role sets we extracted contained exactly such configurations. Further inspection revealed that these irregularities resulted from either noise introduced by errors in the automatic assignment of grammatical functions, or instances with syntactically non-local role assignments.

Starting from such observations, our approach supported a semi-automatic, incremental refinement of the linguistic and annotation models, in this case introducing a distinction between local and non-local role realisations.

Size of the lexicon. Using a series of SeRQL queries, we have computed the size of the corpus/lexicon model for the SALSA/TIGER data (see Table 3). The lexicon model architecture as described in Section 3 results in a total of more than 304,000 instances in the lexicon, instantiating 581 different frame classes and 1,494 role classes.

Type	*No. of instances*
Lemmas	523
Lemma-frame pairs (LUs)	1,176
Sentences	13,353
Syntactic units	223,302
Single-word targets	16,268
Multi-word targets	258
Frame annotations	16,526
Simple	14,700
Underspecified	995
Metaphoric	785
Elliptic	107
Role annotations	31,704
Simple	31,112
Underspecified	592

Table 3: Instance count based on the first SALSA release

5 Consistency Control

The first problem pointed out in Section 1 was the need for efficient consistency control mechanisms. Our OWL DL-based model in fact offers two mechanisms for consistency checking: axiom-based and query-based checking.

Axiom-based checking. Once some constraint has been determined to be universally applicable, it can be formulated in Description Logics in the form of *axiomatic expressions* on the respective class in the model. Although the general interpretation of these axioms in DL is that they allow for inference of new statements, they can still be used as a kind of well-formedness "constraint". For example, if an individual is asserted as an instance of a particular class, the reasoner will detect an inconsistency if this instance does not adhere to the axiomatic class definition. For semantic role annotations, axioms can e.g. define the admissible relations between a particular frame and its roles. This is illustrated in the DL statements below, which express that an instance of PLACING may *at most* have the roles GOAL, PATH, etc.

$$\mathsf{Placing} \sqsubseteq \exists.\mathsf{hasRole}\ (\mathsf{Placing.Goal} \sqcup \mathsf{Placing.Path} \sqcup \ldots)$$
$$\mathsf{Placing} \sqsubseteq \forall.\mathsf{hasRole}\ (\mathsf{Placing.Goal} \sqcup \mathsf{Placing.Path} \sqcup \ldots)$$

Relations between roles can be formalised in a similar way. An example is the *excludes* relation in FrameNet, which prohibits the co-occurrence of roles like CAUSE and AGENT of the PLACING frame. This can be expressed by the following statement.

$$\mathsf{Placing} \sqsubseteq \neg((\exists.\mathsf{hasRole\ Placing.Cause}) \sqcap (\exists.\mathsf{hasRole\ Placing.Agent}))$$

The restrictions are used in checking the consistency of the semantic annotation; violations of these constraints lead to inconsistencies that can be identified by theorem provers. Although current state-of-the-art reasoners do not yet scale to the size of entire corpora, axiom-based checking still works well for our data due to SALSA's policy of dividing the original TIGER corpus into separate subcorpora, each dealing with one particular lemma (cf. Scheffczyk et al. (2006)).

Query-based checking. Due to the nature of our graph representation, constraints can combine different types of information to control adherence to annotation guidelines. Examples are the assignment of the SUPPORTED role of support verb constructions, which ought to be assigned to the maximal syntactic constituent projected by the supported noun, or the exclusion of reflexive pronouns from the span of the target verb. However, the consistency of multi-level annotation is often difficult to check: Not only are some types of classification (e.g. assignment of semantic classes) inherently difficult; the annotations also need to be considered in context. For such cases, axiom-based checking is too strict. In practice, it is important that manual effort can be reduced by automatically extracting subsets of "suspicious" data for inspection. This can be done using SeRQL queries which – in contrast to the general remarks on the scalability of reasoners – are processed and evaluated very quickly on the entire annotated corpus data.

Example queries that we formulated examine suspicious configurations of annotation types, such as target words evoking two or more frame annotations which are neither marked as underspecified nor tagged as a pair of (non-)literal metaphorical frame annotations. Here, we identified 8 cases of omitted annotation markup, namely 4 missing metaphor flags and 4 omitted underspecification links.

On the semantic level, we extracted annotation instances (in context) for metaphorical vs. non-metaphorical readings, or frames that are involved in underspecification in certain sentences, but not in others. While the result sets thus obtained still require manual inspection, they clearly illustrate how the detection of inconsistencies can be enhanced by a declarative formalisation of the annotation scheme. Another strategy could be to concentrate on frames or lemmas exhibiting proportionally high variation in annotation (Dickinson and Meurers, 2003).

6 Conclusion

In this paper, we have constructed a Description Logics-based lexicon model directly from multi-layer linguistic corpus annotations. We have shown how such a model allows for explicit data modelling, and for flexible and fine-grained definition of various degrees of abstractions over corpus annotations.

Furthermore, we have demonstrated that a powerful logical formalisation which integrates an underlying annotation scheme can be used to directly control consistency of the annotations using general KR techniques. It can also overcome limitations of current XML-based search tools by supporting queries which are able to connect multiple levels of linguistic analysis. These queries can be used variously as an additional means of consistency control, to derive quantitative tendencies from the data, to extract lexicon views tailored to specific purposes, and finally as a general tool for linguistic research.

Acknowledgements

This work has been partly funded by the German Research Foundation DFG (grant PI 154/9-2). We also thank the two anonymous reviewers for their valuable comments and suggestions.

References

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. CUP.

Olga Babko-Malaya, Ann Bies, Ann Taylor, Szuting Yi, Martha Palmer, Mitch Marcus, Seth Kulick, and Libin Shen. 2006. Issues in Synchronizing the English Treebank and PropBank. In *Proceedings of the COLING/ACL Workshop on Frontiers in Linguistically Annotated Corpora*, Sydney.

Hans C. Boas. 2005. Semantic frames as interlingual representations for multilingual lexical databases. *International Journal of Lexicography*, 18(4):445–478.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.

Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. 2002. Sesame: A generic architecture for storing and querying RDF and RDF Schema. In *Proceedings of the 1st ISWC*, Sardinia.

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. 2006. The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of the 5th LREC*, Genoa.

Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the 10th EACL*, Budapest.

Charles J. Fillmore, Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235–250.

Gil Francopoulo, Monte George, Nicoletta Calzolari, Monica Monachini, Nuria Bel, Mandy Pet, and Claudia Soria. 2006. LMF for multilingual, specialized lexicons. In *Proceedings of the 5th LREC*, Genoa.

Anette Frank. 2004. Generalisations over corpus-induced frame assignment rules. In *Proceedings of the LREC Workshop on Building Lexical Resources From Semantically Annotated Corpora*, Lisbon.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Catherine Lai and Steven Bird. 2004. Querying and updating treebanks: A critical survey and requirements analysis. In *Proceedings of the Australasian Language Technology Workshop*, Sydney.

Christophe Laprun, Jonathan Fiscus, John Garofolo, and Sylvain Pajot. 2002. Recent Improvements to the ATLAS Architecture. In *Proceedings of HLT 2002*, San Diego.

Eleni Miltsakaki, Nikhil Dinesh, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2005. Experiments on sense annotations and sense disambiguation of discourse connectives. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, Barcelona, Spain.

Alain Polguère. 2006. Structural properties of lexical systems: Monolingual and multilingual perspectives. In *Proceedings of the COLING/ACL Workshop on Multilingual Language Resources and Interoperability*, Sydney.

Jan Scheffczyk, Collin F. Baker, and Srini Narayanan. 2006. Ontology-based reasoning about lexical resources. In *Proceedings of the 5th OntoLex*, Genoa.

Constructing Taxonomy of Numerative Classifiers for Asian Languages

Kiyoaki Shirai
JAIST
kshirai@jaist.ac.jp

Takenobu Tokunaga
Tokyo Inst. of Tech.
take@cl.cs.titech.ac.jp

Chu-Ren Huang
Academia Sinica
churenhuang@gmail.com

Shu-Kai Hsieh
National Taiwan Normal Univ.
shukai@gmail.com

Tzu-Yi Kuo
Academia Sinica
ivykuo@gate.sinica.edu.tw

Virach Sornlertlamvanich
TCL, NICT
virach@tcllab.org

Thatsanee Charoenporn
TCL, NICT
thatsanee@tcllab.org

Abstract

Numerative classifiers are ubiquitous in many Asian languages. This paper proposes a method to construct a taxonomy of numerative classifiers based on a noun-classifier agreement database. The taxonomy defines superordinate-subordinate relation among numerative classifiers and represents the relations in tree structures. The experiments to construct taxonomies were conducted for evaluation by using data from three different languages: Chinese, Japanese and Thai. We found that our method was promising for Chinese and Japanese, but inappropriate for Thai. It confirms that there really is no hierarchy among Thai classifiers.

1 Introduction

Many Asian languages do not mark grammatical numbers (singular/plural) in noun form, but use numerative classifiers together with numerals instead when describing the number of nouns. Numerative classifiers (hereafter "classifiers") are used with a limited group of nouns, in particular material nouns. In English, for example: "three pieces of paper". In Asian languages these classifiers are ubiquitous and used with common nouns. Therefore the number of classifiers is much larger than in Western languages. An agreement between nouns and classifiers is also necessary, i.e., a certain noun specifies possible classifiers. The agreement is determined based on various aspects of a noun, such as its meaning, shape, pragmatic aspect and so on.

This paper proposes a method to automatically construct a taxonomy of numerative classifiers for Asian languages. The taxonomy defines superordinate-subordinate relations between classifiers. For instance, the Japanese classifier "頭 (*tô*)" is used for counting big animals such as elephants and tigers, while "匹 (*hiki*)" is used for all animals. Since "匹" can be considered more general than "頭", "匹" is the superordinate classifier of "頭", represented as "匹" $\succ$ "頭" in this paper. The taxonomy represents such superordinate-subordinate relations between classifiers in the form of a tree structure. A taxonomy of classifiers would be fundamental knowledge for natural language processing. In addition, it will be useful for language learners, because learning usage of classifiers is rather difficult, especially for Western language speakers.

We evaluate the proposed method by using the data of three Asian languages: Chinese, Japanese and Thai.

2 Noun-classifier agreement database

First, let us introduce usages of classifiers in Asian languages. In the following examples, "CL" stands for classifier.

- Chinese: *yi-ju* (CL) *dian-hua* (telephone) $\cdots$ a telephone
- Japanese: *inu* (dog) *2* *hiki* (CL) $\cdots$ 2 dogs
- Thai: *nakrian* (student) *3* *khon* (CL) $\cdots$ 3 students

As mentioned earlier, the agreement between nouns and classifiers is observed. For instance, the Japanese classifier "*hiki*" in the above example agrees with only animals. The agreement is also found in Chinese and Thai.

The proposed method to construct a classifier taxonomy is based on agreement between nouns and classifiers. First we prepare a collection of pairs (n, c) of a noun n and a classifier c which agrees with n for a language. The statistics of our Chinese, Japanese, and Thai database are summarized in Table 1.

Table 1: Noun-classifier agreement database

	Chinese	Japanese	Thai
No. of (n,c) pairs	28,202	9,582	9,618
No. of nouns (type)	10,250	4,624	8,224
No. of CLs (type)	205	331	608

The Japanese database was built by extracting noun-classifier pairs from a dictionary (Iida, 2004) which enumerates nouns and their corresponding classifiers. The Chinese database was derived from a dictionary (Huang et al., 1997). The Thai database consists of a mixture of two kinds of noun-classifier pairs: 8,024 nouns and their corresponding classifiers from a dictionary of a machine translation system (CICC, 1995) and 200 from a corpus. The pairs from the corpus were manually checked for their validity.

3 Proposed Method

3.1 Extracting superordinate-subordinate relations of classifiers

We extracted superordinate-subordinate classifier pairs based on inclusive relations of sets of nouns agreeing with those classifiers. Suppose that N_k is a set of nouns that agrees with a classifier c_k. If N_i subsumes N_j ($N_i \supset N_j$), we can estimate that c_i subsumes c_j ($c_i \succ c_j$). For instance, in our Japanese database, the classifier "店 (*ten*)" agrees with shops such as "drug store", "kiosk" and "restaurant", and these nouns also agree with "軒 (*ken*)", since "軒" is a classifier which agrees with any kind of building. Thus, we can estimate the relation "軒" $\succ$ "店".

Given a certain classifier c_j, c_i satisfying the following two conditions (1) and (2) is considered as a superordinate classifier of c_j.

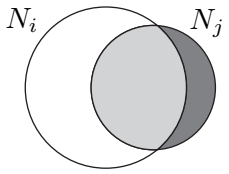


Figure 1: Relation of sets of nouns agreeing with classifiers

$$|N_i| > |N_j| \quad (1)$$

$$\mathrm{IR}(c_i, c_j) \geq T_{ir} \quad (2)$$

$$\text{where } \mathrm{IR}(c_i, c_j) \stackrel{def}{=} \frac{|N_i \cap N_j|}{|N_j|}$$

Condition (1) requires that a superordinate classifier agrees with more nouns than a subordinate classifier. $\mathrm{IR}(c_i, c_j)$ is an inclusion ratio representing to what extent nouns in N_j are also included in N_i (the ratio of the light gray area to the area of the small circle in Figure 1).

Condition (2) means that if $\mathrm{IR}(c_i, c_j)$ is greater than a certain threshold T_{ir}, we estimate a superordinate-subordinate relation between c_i and c_j. The basic idea is that superordinate-subordinate relations are extracted when N_j is a proper subset of N_i, i.e. $\mathrm{IR}(c_i, c_j) = 1$, but this is too strict. In order to extract more relations, we loosen this condition such that relations are extracted when $\mathrm{IR}(c_i, c_j)$ is large enough. If we set T_{ir} lower, more relations can be acquired, but they may be less reliable.

Table 2: Extraction of superordinate-subordinate relations

	Chinese	Japanese	Thai
T_{ir}	0.7	0.6	0.6
No. of extracted relations	251	322	239
No. of CLs not in the extracted relations	36 (18%)	76 (23%)	395 (61%)

Table 2 shows the results of our experiments to extract superordinate-subordinate relations of classifiers. The threshold T_{ir} was determined in an *ad hoc* manner for each language. The numbers of extracted superordinate-subordinate relations are shown in the second row in the table. Manual inspection of the sampled relations revealed that many reasonable relations were extracted. The objective evaluation of these extracted relations will be discussed in 4.2.

The third row in Table 2 indicates the numbers of classifiers which were not included in the extracted superordinate-subordinate relations with its ratio to the total number of classifiers in the database in parentheses. We found that no relation is extracted for a large number of Thai classifiers.

3.2 Constructing structure

The structure of a taxonomy is constructed based on a set of superordinate-subordinate relations between classifiers. Currently we adopt a very naive approach to construct structures, i.e., starting from the most superordinate classifiers as roots, we extend trees downward to less general classifiers by using the extracted superordinate-subordinate relations. Note that since there is more than one classifier that does not have any superordinate classifiers, we will have a set of trees rather than a single tree.

When constructing structures, redundant relations are ignored in order to make the structures as concise as possible. A relation is considered redundant if the relation can be inferred by using other relations and transitivity of the relations. The formal definition of redundant relations is given below:

$$c_a \succ c_b \text{ is redundant } \textit{iff } \exists c_m : c_a \succ c_m, c_m \succ c_b$$

Statistics of constructed structures for each language are shown in Table 3. More than 50 isolated structures (trees) were obtained for Chinese and Japanese, while more than 100 for Thai. We obtained several large structures, the largest containing 45, 85 and 23 classifiers for Chinese, Japanese and Thai, respectively. As indicated in the fifth row in Table 3, however, many structures consisting of only 2 classifiers were also constructed.

Table 3: Construction of structures

	Chinese	Japanese	Thai
No. of structures	52	54	102
No. of CLs in a structure			
Average	4.9	6.3	3.3
Maximum	45	85	23
Max. depth of structures	4	3	3
No. of structures with 2 CLs	18	24	54

4 Discussion

In this section, we will discuss the results of our experiments. First 4.1 discusses appropriateness of our method for the three languages. Then we evaluate our method in more detail. The evaluation of extracted superordinate-subordinate relations is described in 4.2, and the evaluation of structures in 4.3.

4.1 Comparison of different languages

According to the results of our experiments, the proposed method seems promising for Chinese and Japanese, but not for Thai. From the Thai data, no relation was obtained for about 60% of classifiers (Table 2), and many small fragmented structures were created (Table 3).

This is because of the characteristic that nouns and classifiers are strongly coupled in Thai, i.e., many classifiers agree with only one noun. In our Thai database, 252 (41.5%) classifiers agree with only one noun. This means that the overlap between two noun sets N_i and N_j can be quite small, making the inclusion ratio $\mathrm{IR}(c_i, c_j)$ very small. Out basic idea is that we can extract superordinate-subordinate relations between two classifiers when the overlap of their corresponding noun sets is large. However, this assumption does not hold in Thai classifiers. The above facts suggest that there seems to be no hierarchical taxonomy of classifiers in Thai.

4.2 Evaluation of extracted relations

4.2.1 Analysis of Nouns in $N_j \setminus N_i$

As explained in 3.1, our method extracts a relation $c_i \succ c_j$ even when N_i does not completely subsume N_j. We analysed nouns in the relative complement of N_i in N_j ($N_j \setminus N_i$), i.e., the dark gray area in Figure 1. The relation $c_i \succ c_j$ implies that all nouns which are countable with a subordinate classifier c_j are also countable with its superordinate classifier c_i, but there is no guarantee of this for nouns in $N_j \setminus N_i$, since we loosened the condition as in (2) by introducing a threshold.

To see to what extent nouns in $N_j \setminus N_i$ agree with c_i as well, we manually verified the agreement of nouns in $N_j \setminus N_i$ and c_i for all extracted relations $c_i \succ c_j$. The verification was done by native speakers of each language. Results of the validation are summarized in Table 4. For Japanese and Chinese, multiple judges verified the results. When judgments conflicted, we decided the final decision by a discussion of two judges for Japanese, and by majority voting for Chinese. The 4th and 5th rows

in Table 4 show the agreement of judgments. The "Agreement ratio" is the ratio of cases that judgments agree. Since three judges verified nouns for Chinese, we show the average of the agreement ratios for two judges out of the three. The agreement ratio and Cohen's κ is relatively high for Japanese, but not for Chinese. We found many uncertain cases for Chinese nouns. For example, "位 (*wei*)" is a classifier used when counting people with honorific perspective. However, judgement if "位" can modify nouns such as "political prisoner" or "local villain" is rather uncertain.

Table 4: Analysis of nouns in $N_j \setminus N_i$

	Chinese	Japanese	Thai
No. of nouns in $N_j \setminus N_i$	1,650	579	43
No. of nouns countable	1,195	241	24
with c_i as well	72%	42%	56%
No. of judges	3	2	1
Agreement ratio	0.677	0.936	–
Cohen's κ	0.484	0.868	–

Table 4 reveals that a considerable number of nouns in $N_j \setminus N_i$ are actually countable with c_i, meaning that our databases do not include noun-classifier agreement exhaustively.

4.2.2 Reliability of relations "$\succ$"

Based on the analysis in 4.2.1, we evaluate extracted superordinate-subordinate relations. We define the reliability R of the relation $c_i \succ c_j$ as

$$\mathrm{R}(c_i \succ c_j) \quad = \quad \frac{|N_i \cap N_j| + |NC_{j,i}|}{|N_j|}, \qquad (3)$$

where, $NC_{j,i}$ is a subset of $N_j \setminus N_i$ consisting of nouns which are manually judged to agree with c_i. We can consider that the more strictly this statement holds, the more reliable the extracted relations will be.

Figure 2 shows the relations between the threshold T_{ir} and both the number of extracted relations and their reliability. The horizontal axis indicates the threshold T_{ir} in (2). The bar charts indicate the number of extracted relations, while the line graphs indicate the averages of reliability of all extracted relations. Of course, if we set T_{ir} lower, we can extract more relations at the cost of their reliability. However, even when T_{ir} is set to the lowest value, the averages of reliability are relatively high, i.e. 0.98 (Chinese), 0.91 (Japanese) and 0.99 (Thai). Thus we can conclude that the extracted superordinate-subordinate relations are reliable enough.

4.3 Evaluation of structures

As in ordinary ontologies, we will assume that properties of superordinate classifiers can be inherited to their subordinate classifiers. In other words, a classifier taxonomy suggests transitivity of agreement with nouns over superordinate-subordinate relations as

$$c_1 \succ c_2 \ \wedge \ c_2 \succ c_3 \quad \Rightarrow \quad c_1 \succ c_3.$$

In order to evaluate the structures of our taxonomy, we verify the validity of transitivity.

First, we extracted all pairs of classifiers having an ancestor-descendant relation from our classifier taxonomy. Hereafter we denote ancestor-descendant pairs of classifiers as (c_a, c_d), where c_a is an ancestor and c_d an descendant. The path from c_a to c_d on the taxonomy can be represented as

$$c_0(= c_a) \succ c_1 \succ ... \succ c_n(= c_d). \qquad (4)$$

We denote a superordinate-subordinate relation derived by transitivity as $\overset{*}{\succ}$, such as $c_0 \overset{*}{\succ} c_n$. Among all ancestor-descendant relations, we extracted ones with a path length of more than one, or $n > 1$ in (4). Then we compare $\mathrm{R}(c_a \overset{*}{\succ} c_d)$, the reliability of a relation derived by transitivity, with $\mathrm{R}(c_i \succ c_{i+1})$ $(0 \leq i < n)$, the reliability of direct relations in the path from c_a to c_d. If these are comparable, we can conclude that transitivity in the taxonomy is valid.

Table 5 shows the results of the analysis of transitivity. As indicated in the column "all" in Table 5, 78 and 86 ancestor-descendant pairs (c_a, c_d) were extracted from the Chinese and Japanese classifier taxonomy, respectively. In contrast, only 6 pairs were extracted from the Thai taxonomy, since each structure of the Thai taxonomy is rather small as we already discussed with Table 3. Thus we have omitted further analysis of Thai. The extracted ancestor-descendant pairs of classifiers are then classified into three cases, (A), (B) and (C). Their numbers are shown in the last three rows in Table 5, where $\min_i$ and $\max_i$ denote the minimum and maximum of reliability among all direct relations $\mathrm{R}(c_i \succ c_{i+1})$ in the path from c_a to c_d.

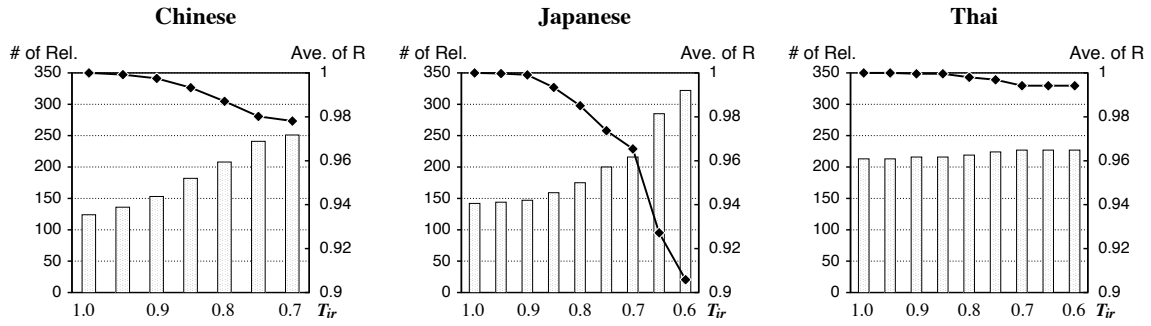


Figure 2: Reliability of extracted superordinate-subordinate relations

Table 5: Verification of transitivity

	Chinese			Japanese		
	all	direct	indirect	all	direct	indirect
No. of (c_a, c_d)	78	58	20	86	55	31
Average of $\mathrm{R}(c_a \overset{*}{\succ} c_d)$	0.88	0.98	0.61	0.77	0.93	0.48
(A) $\min_i > \mathrm{R}(c_a \overset{*}{\succ} c_d)$	16 (21%)	4 (7%)	12 (60%)	24 (28%)	3 (5%)	21 (68%)
(B) $\min_i \leq \mathrm{R}(c_a \overset{*}{\succ} c_d) < \max_i$	39 (50%)	34 (59%)	5 (25%)	27 (31%)	24 (44%)	3 (9%)
(C) $\max_i \leq \mathrm{R}(c_a \overset{*}{\succ} c_d)$	23 (29%)	20 (34%)	3 (15%)	35 (41%)	28 (51%)	7 (23%)

In case (A), reliability of a relation derived by transitivity, $\mathrm{R}(c_a \overset{*}{\succ} c_d)$, is less than that of any direct relations, $\mathrm{R}(c_i \succ c_{i+1})$. In case (B), reliability of a transitive relation is comparable with that of direct relations, i.e. $\mathrm{R}(c_a \overset{*}{\succ} c_d)$ is greater or equal to min_i and less than max_i. In case (C), the transitive relation is more reliable than direct relations.

The average of the reliability of $c_a \overset{*}{\succ} c_d$ is relatively high, 0.88 for Chinese and 0.77 for Japanese. We also found that more than 70% of derived relations (case (B) and case (C)) are comparable to or greater than direct relations. The above facts indicate transitivity on our structural taxonomy is valid to some degree.

From a different point of view, we divided pairs of (c_a, c_d) into two other cases, "direct" and "indirect" as shown in the columns of Table 5. The "direct" case includes the relations which are also extracted by our method. Note that such relations are discarded as redundant ones. On the other hand, the "indirect" case includes the relations which can not be extracted from the database but only inferred by using transitivity on the taxonomy. That is, they are truly new relations. In order to calculate reliability of "indirect" cases, we performed additional manual validation of nouns in $N_d \backslash N_a$.

However, the average of $\mathrm{R}(c_a \overset{*}{\succ} c_d)$ in "indirect" cases is not so high for both Chinese and Japanese, as a large amount of pairs are classified into case (A). Thus it is not effective to infer new superordinate-subordinate relations by transitivity. Since we currently only adopted a very naive method to construct a classifier taxonomy, more sophisticated methods should be explored in order to prevent inferring irrelevant relations.

5 Related Work

Bond (2000) proposed a method to choose an appropriate classifier for a noun by referring its semantic class. This method is implemented in a sentence generation module of a machine translation system. Similar attempts to generate both Japanese and Korean classifiers were also reported (Paik and Bond, 2001). Bender and Siegel (2004) implemented a HPSG that handles several intricate structures including Japanese classifiers. Matsumoto (1993) reported his close analysis of Japanese classifiers based on prototype semantics. Sornlertlamvanich (1994) presented an algorithm for selecting an adequate classifier for a noun by using a corpus. Their research can be regarded as a method to construct a noun-classifier agreement database au-

tomatically from corpora. We used databases derived from dictionaries except for a small number of noun-classifier pairs in Thai, because we believe dictionaries provide more reliable and stable information than corpora, and in addition they were available and on hand. Note that we are not concerned with frequencies of noun-classifier coocurrence in this study. Huang (1998) proposed a method to construct a noun taxonomy based on noun-classifier agreement that is very similar to ours, but aims at developing a taxonomy for nouns rather than one for classifiers. There has not been very much work on building resources concerning noun-classifier agreement. To our knowledge, this is the first attempt to construct a classifier taxonomy.

6 Conclusion

This paper proposed a method to construct a taxonomy of numerative classifiers based on a noun-classifier agreement database. First, superordinate-subordinate relations of two classifiers are extracted by measuring the overlap of two sets of nouns agreeing with each classifier. Then these relations are used as building blocks to build a taxonomy of tree structures. We conducted experiments to build classifier taxonomies for three languages: Chinese, Japanese and Thai. The effectiveness of our method was evaluated by measuring reliability of extracted relations, and verifying validity of transitivity in the taxonomy. We found that extracted relations are reliable, and the transitivity in the taxonomy relatively valid. Relations inferred by transitivity, however, are less reliable than those directly derived from noun-classifier agreement.

Future work includes investigating a way to enlarge classifier taxonomies. Currently, not all classifiers are included in our taxonomy, and it consists of a set of fragmented structures. A more sophisticated method to build a large taxonomy including more classifiers should be examined. Our method should also be refined in order to make superordinate-subordinate relations inferred by the transitivity more reliable. We are now investigating a stepwise method to construct taxonomies that prefers more reliable relations, i.e. an initial taxonomy is built with a small number of highly reliable relations, and is then expanded with less reliable ones.

Acknowledgment

This research was carried out through financial support provided under the NEDO International Joint Research Grant Program (NEDO Grant).

References

Emily M. Bender and Melanie Siegel. 2004. Implementing the syntax of Japanese numeral classifiers. In *Proceedings of the the First International Joint Conference on Natural Language Processing*, pages 398–405.

Francis Bond and Kyonghee Paik. 2000. Reusing an ontology to generate numeral classifiers. In *Proceedings of the COLING*, pages 90–96.

CICC. 1995. CICC Thai basic dictionary. (developed by Center of the International Cooperation for Computerization).

Chu-Ren Huang, Keh-Jian Chen, and Chin-Hsiung Lai, editors. 1997. *Mandarin Daily News Dictionary of Measure Words*. Mandarin Daily News Publisher.

Chu-Ren Huang, Keh-jiann Chen, and Zhao-ming Gao. 1998. Noun class extraction from a corpus-based collocation dictionary: An integration of computational and qualitative approaches. In *Quantitative and Computational Studies of Chinese Linguistics*, pages 339–352.

Asako Iida. 2004. *Kazoekata no Ziten (Dictionary for counting things)*. Shôgakukan. (in Japanese).

Yo Matsumoto. 1993. The Japanese numeral classifiers: A study of semantic categories and lexical organization. *Linguistics*, 31:667–713.

Kyonghee Paik and Francis Bond. 2001. Multilingual generation of numeral classifiers using a common ontology. In *Proceedings of the 19th International Conference on Computer Processing of Oriental Languages (ICCPOL)*, pages 141–147.

Virach Sornlertlamvanich, Wantanee Pantachat, and Surapant Meknavin. 1994. Classifier assignment by corpus-based approach. In *Proceedings of the COLING*, pages 556–561.

Translating Compounds by Learning Component Gloss Translation Models via Multiple Languages

Nikesh Garera and **David Yarowsky**
Department of Computer Science
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218, USA
{ngarera,yarowsky}@cs.jhu.edu

Abstract

This paper presents an approach to the translation of compound words without the need for bilingual training text, by modeling the mapping of literal component word glosses (e.g. "iron-path") into fluent English (e.g. "railway") across multiple languages. Performance is improved by adding component-sequence and learned-morphology models along with context similarity from monolingual text and optional combination with traditional bilingual-text-based translation discovery.

1 Introduction

Compound words such as *lighthouse* and *fireplace* are words that are composed of two or more component words and are often a challenge for machine translation due to their potentially complex compounding behavior and ambiguous interpretations (Rackow et al., 1992). For many languages, such words form a significant portion of the lexicon and the compounding process is further complicated by diverse morphological processes (Levi, 1978) and the properties of different compound sequences such as Noun-Noun, Adj-Adj, Adj-Noun, Verb-Verb, etc. Compounds also tend to have a high type frequency but a low token frequency which makes their translation difficult to learn using corpus-based algorithms (Tanaka and Baldwin, 2003). Furthermore, most of the literature on compound translation has been restricted to a few languages dealing with compounding phenomena specific to the language in question.

Compound	Splitting	English Gloss	Translation
Input: Distilled glosses from German-English dictionary			
Krankenhaus	Kranken-Haus	sick-house	hospital
Regenschirm	Regen-Schirm	rain-guard	umbrella
WörterBuch	Wörter-Buch	words-book	dictionary
Eisenbahn	Eisen-Bahn	**iron-path**	railroad
Input: Distilled glosses from Swedish-English dictionary			
Sjukhus	Sjhu-Khus	sick-house	hospital
Järnväg	Järn-väg	**iron-path**	railway
Ordbok	Ord-Bok	words-book	dictionary
Goal: To translate new Albanian compounds			
Hekurudhë	Hekur-Udhë	**iron-path**	**???**

Table 1: Example lexical resources used in this task and their application to translating compound words in new languages.

With these challenges in mind, the primary goal of this work is to improve the coverage of translation lexicons for compounds, as illustrated in Table 1 and Figure 1, in multiple new languages. We show how using cross-language compound evidence obtained from bilingual dictionaries can aid in compound translation. A primary motivating idea for this work is that the literal component glosses for compound words (such as "iron path" for *railway*) is often replicated in multiple languages, providing insight into the fluent translation of a similar literal gloss in a new (often resource-poor) language.

2 Resources Utilized

The only resource utilized for our compound translation lexicon algorithm is a collection of bilingual dictionaries. We used bilingual dictionary collections for 50 languages that were acquired in electronic form over the Internet or via optical character recognition (OCR) on paper dictionaries. Note that *no parallel or even monolingual corpora is required*, their use described later in the paper is optional.

3 Related Work

The compound-translation literature typically deals with these steps: 1) Compound splitting, 2) translation candidate generation and 3) translation candidate scoring. Compound splitting is generally done using translation lexicon lookup and allowing for different splitting options based on corpus frequency (Zhang et al., 2000; Koehn and Knight, 2003).

Translation candidate generation is an important phase and this is where our work differs significantly from the previous literature. Most of the previous work has been focused on generating *compositional* translation candidates, that is, the translation candidates of the compound words are lexically composed of the component word translations. This has been done by either just concatenating the translations of component words to form a candidate (Grefenstette, 1999; Cao and Li, 2002), or using syntactic templates such as "E_2 in E_1", "E_1 of E_2" to form translation candidates from the translation of the component words E_2 and E_1 (Baldwin and Tanaka, 2004), or using synsets of the component word translations to include synonyms in the compositional candidates (Navigli et al., 2003).

The above class of work in compositional-candidate generation fails to translate compounds such as *Krankenhaus (hospital)* whose component word translations are *Kranken (sick) and Haus (hospital)*, and composing *sick* and *house* in any order will not result in the correct translation (*hospital*). Another problem with using fixed syntactic templates is that they are restricted to the specific patterns occurring in the target language. We show how one can use the gloss patterns of compounds in multiple other languages to hypothesize translation candidates that are not lexically compositional.

4 Approach

Our approach to compound word translation is illustrated in Figure 1.

4.1 Splitting compound words and gloss generation with translation lexicon lookup

We first split a given source word, such as the Albanian compound *hekurudhë*, into a set of component word partitions, such as *hekur (iron)* and *udhë (path)*. Our initial approach is to consider all possible partitions based on contiguous component words found in a small dictionary for the language, as in

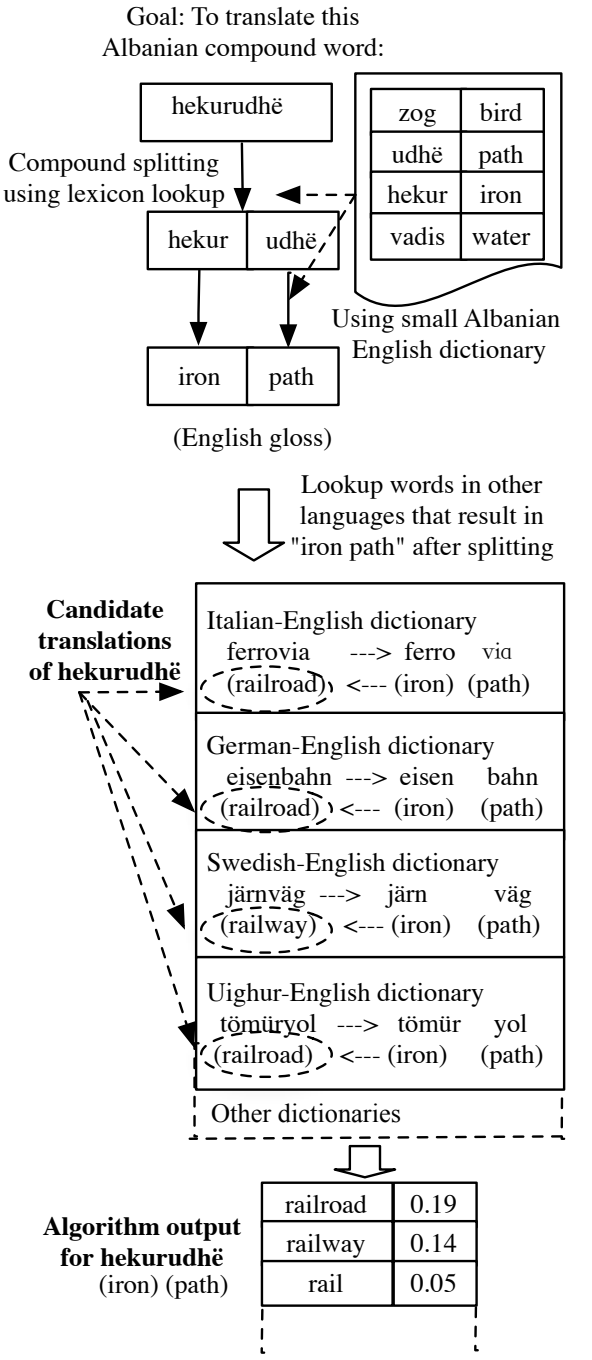


Figure 1: Illustration of using cross-language evidence using bilingual dictionaries of different languages for compound translation

Brown (2002) and Koehn and Knight (2003)[1]. For a given split, we generate its English glosses by using all possible English translations of the component words given in the dictionary of that language[2].

4.2 Using cross-language evidence from different bilingual dictionaries

For many compound words (especially for borrowings), the compounding process is identical across several languages and the literal English gloss remains the same across these languages. For example, the English word *railway* is translated as a compound word in many languages, and the English gloss of those compounds is often *"iron path"* or a similar literal meaning[3]. Thus knowing the fluent English translation of the literal gloss *"iron path"* in some relatively resource-rich language provides a vehicle for the translation from all other languages sharing that literal gloss[4]

4.3 Ranking translation candidates

The confidence in the correctness of a mapping between a literal gloss (e.g. "iron path") and fluent translation (e.g. "railroad") can be based on the number of distinct languages exhibiting this association. Thus we rank the candidate translations generated via different languages as in Figure 1 as follows: For a given target compound word, say f_c with a set of English glosses G obtained via multiple splitting options or multiple component word translations, the translation probability for a candidate translation can be computed as:

$$\begin{aligned} p(e_c|f_c) &= \sum_{g \in G} p(e_c, g|f_c) \\ &= \sum_{g \in G} p(g|f_c) \cdot p(e_c|g, f_c) \\ &= \sum_{g \in G} p(g|f_c) \cdot p(e_c|g) \end{aligned}$$

where, $p(g|f_c) = p(g_1|f_1) \cdot p(g_2|f_2)$. f_1, f_2 are the individual component-words of compound and g_1, g_2 are their translations from the existing dictionary. For human dictionaries, $p(g|f_c)$ is uniform for all $g \in G$, while variable probabilities can also be acquired from bitext or other translation discovery approaches. Also, $p(e_c|g) = \frac{freq(g,e_c)}{freq(g)}$, where $freq(g, e_c)$ is the number of times the compound word with English gloss g is translated as e_c in the bilingual dictionaries of *other* languages and $freq(g)$ is the total number of times the English gloss appears in these dictionaries.

5 Evaluation using Exact-match Translation Accuracy

For evaluation, we assess the performance of the algorithm on the following 10 languages: Albanian, Arabic, Bulgarian, Czech, Farsi, German, Hungarian, Russian, Slovak and Swedish. We detail both the average performance for these 10 languages (Avg_{10}), as well as provide individual performance details on Albanian, Bulgarian, German and Swedish. For each of the compound translation models, we report coverage (the # of compound words for which a hypothesis was generated by the algorithm) and Top1/Top10 accuracy. Top1 and Top 10 accuracy are the fraction of words for which a correct translation (listed in the evaluation dictionary) appears in the Top 1 and Top 10 translation candidates respectively, as ranked by the algorithm. Because evaluation dictionaries are often missing acceptable translations (e.g. *railroad* rather than *railway*), and any deviation from exact-match is scored as incorrect, these measures will be a lower bound on acceptable translation accuracy. Also, target language models can often select effectively among such hypothesis lists in context.

6 Comparison of different compound translation models

6.1 A simple model using literal English gloss concatenation as the translation

Our baseline model is a simple gloss concatenation model for generating compositional translation candidates on the lines of Grefenstette (1999) and Cao and Li (2002). We take the translations of the individual component-words (e.g. for the compound word *hekurudhë*, they would be *hekur (iron)* and

[1] In order to avoid inflections as component-words we limit the component-word length to at least three characters.

[2] The algorithm is allowed to generate multiple glosses "iron way," "iron road," etc. based on multiple translations of the component words. Multiple glosses only add to the number of translation candidates generated.

[3] For the gloss, "iron path", we found 10 other languages in which some compound word has the English gloss after splitting and component-word translation

[4] We do assume an existing small translation lexicon in the target language for the individual component-words, but these are often higher frequency words and present either in a basic dictionary or discoverable through corpus-based techniques.

udhë (path)) and hypothesizes three translation candidate variants: "iron path", "iron-path" and "iron-path". A test instance is scored as correct if any of these translation candidates occur in the translations of *hekurudhë* in the bilingual dictionary. This baseline performance measures how well simple literal glosses serve as translation candidates. In cases such as the German compound *Nußschale* (*nutshell*), which is a simple concatenation of the individual components *Nuß(nut)* and *Schale (shell)*, the literal gloss is correct. For this baseline, if the component-words have multiple translations, then each of the possible English gloss is ranked randomly. While Grefenstette (1999) and Cao and Li (2002) proposed re-ranking these candidates using web-data, the potential gains of this ranking are limited, as we see in Table 2 that even the *Found Acc.* is very low[5], that is for most of the cases the correct translation does not appear anywhere in the set of English glosses[6]

Language	Cmpnd wrds translated	Top1 Acc.	Top10 Acc.	Found Acc.
Albanian	4472 (10.11%)	0.001	0.010	0.020
Bulgarian	9093 (12.50%)	0.001	0.015	0.031
German	15731 (29.11%)	0.004	0.079	0.134
Swedish	18316 (31.57%)	0.005	0.068	0.111
Avg_{10}	14228 (17.84%)	0.002	0.030	0.055

Table 2: Baseline performance using unreordered literal English glosses as translations. The percentages in parentheses indicate what fraction of all the words in the test (entire) vocabulary were detected and translated as compounds.

6.2 Using bilingual dictionaries

This section describes the results from the model explained in Section 4. To recap, this model attempts to translate every test word such that there is at least one additional language whose bilingual dictionary supports an equivalent split and literal English gloss, and bases its translation hypotheses on the consensus fluent translation(s) corresponding to the literal glosses in these other languages. The performance is shown in Table 3. The substantial increase in accuracy over the baseline indicates the usefulness of such gloss-to-translation guidance from other languages. The rest of the sections detail our investigation of improvements to this model.

Language	Compound words translated	Top1 Acc.	Top10 Acc.
Albanian	3085 (6.97%)	0.185	0.332
Bulgarian	6719 (9.24%)	0.247	0.416
German	11103 (20.55%)	0.195	0.362
Swedish	12681 (21.86%)	0.188	0.346
Avg_{10}	9320.9 (11.98%)	0.184	0.326

Table 3: Coverage and accuracy for the standard model using gloss-to-fluent translation mappings learned from bilingual dictionaries in other languages (in forward order only).

6.3 Using forward and backward ordering for English gloss search

In our standard model, the literal English gloss for a source compound word (for example, *iron path*) matches glosses in other language dictionaries only in the identical order. But given that modifier/head word order often differs between languages, we test how searching for both orderings (e.g. *"iron path" and "path iron"*) can improve performance, as shown in Table 4. The percentages in parentheses show relative increase from the performance of the standard model in Section 6.2. We see a substantial improvement in both coverage and accuracy.

Language	Cmpnd wrds translated	Top1 Acc.	Top10 Acc.
Albanian	3229(+4.67%)	.217(+17.30%)	.409(+23.19%)
Bulgarian	6806(+1.29%)	.255(+3.24%)	.442(+6.25%)
German	11346(+2.19%)	.199(+2.05%)	.388(+7.18%)
Swedish	12970(+2.28%)	.189(+0.53%)	.361(+4.34%)
Avg_{10}	9603(+3.03%)	.193(+4.89%)	.362(+11.04%)

Table 4: Performance for looking up English gloss via both orderings. The percentages in parentheses are relative improvements from the performance in Table 3

6.4 Increasing coverage by automatically discovering compound morphology

For many languages, the compounding process introduces its own morphology (Figure 2). For example, in German, the word *Geschäftsführer (manager)* consists of the lexemes *Geschäft (business)* and *Führer (guide)* joined by the lexeme *-s*. For the purposes of these experiments, we will call such lexemes *fillers or middle glue characters*. Koehn and Knight (2003) used a fixed set of two known fillers *s and es* for handling German compounds. To broaden the applicability of this work to new languages without linguistic guidance, we show how such fillers

[5]Found Acc. is the fraction of examples for which the correct translation appears *anywhere* in the n-best list

[6]One explanation for this could be that for only a small percentage of compound words, their dictionary translations are formed by concatenating their English glosses. Also, Grefenstette (1999) reports much higher accuracies for German on this model because the 724 German test compounds were chosen in such a way that their correct translation is a concatenation of the possible component word translations.

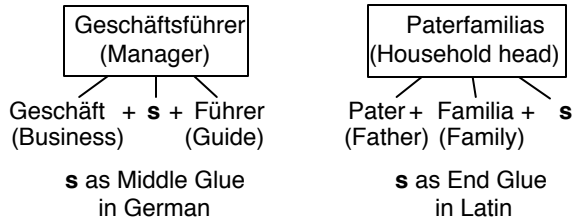


Figure 2: Illustration of compounding morphology using middle and end glue characters.

can be estimated directly from corpora in different languages. In additional to fillers, compound can also introduce morphology at the suffix or prefix of compounds, for example, in the Latin language, the lexeme *paterfamilias* contains the genitive form *familias* of the lexeme *familia (family)*, thus *s* in this case is referred to as the *"end glue"* character. To

Albanian		**Bulgarian**		**German**		**Swedish**	
Top 5 Middle Glue Character(s)							
j	0.059	О	0.129	s	0.133	s	0.132
s	0.048	И	0.046	n	0.090	l	0.051
t	0.042	Н	0.036	k	0.066	n	0.049
r	0.042	Я	0.025	h	0.042	t	0.045
i	0.038	А	0.025	f	0.037	r	0.035
Top 5 End Glue Character(s)							
m	0.146	Т	0.124	n	0.188	a	0.074
t	0.079	ЕН	0.092	t	0.167	g	0.073
s	0.059	Н	0.063	en	0.130	t	0.059
k	0.048	М	0.049	e	0.069	e	0.057
r	0.037	АМ	0.047	d	0.043	d	0.057

Table 5: Top 5 middle glues (fillers) and end glues discovered for each language along with their probability scores.

augment the splitting step outlined in Section 4.1, we allow deletion of up to two middle characters and two end characters. Then, for each glue candidate (for example *es*), we estimate its probability as the relative frequency of unique hypothesized compound words successfully using that particular glue. We rank the set of glues by their probability and take the top 10 middle and end glues for each language. A sample of glues discovered for some of the languages are shown in Table 5. The performance for the morphology step is shown in Table 6. The relative percentage improvements are with respect to the previous Section 6.3. We observe significant gain in coverage as the flexibility of glue process allows discovery of more compounds.

6.5 Re-ranking using context vector projection

We may further improve performance by re-ranking candidate translations based on the goodness of semantic "fit" between two words, as measured by

Language	**Cmpnd wrds translated**	**Top1 Acc.**	**Top10 Acc.**
Albanian	3272(+1.33%)	.214(-1.38%)	.407(-0.49%)
Bulgarian	7211(+5.95%)	.258(+1.18%)	.443(+0.23%)
German	13372(+17.86%)	.200(+0.50%)	.391(+0.77%)
Swedish	15094(+16.38%)	.190(+0.53%)	.363(+0.55%)
Avg_{10}	10273(+6.98%)	.194(+0.52%)	.363(+0.28%)

Table 6: Performance for increasing coverage by including compounding morphology. The percentages in parentheses are relative improvements from the performance in Table 4

their context similarity. This can be accomplished as in Rapp (1999) and Schafer and Yarowsky (2002) by creating bag-of-words context vectors around both the source and target language words and then projecting the source vectors into the (English) target space via the current small translation dictionary. Once in the same language space, source words and their translation hypotheses are compared via cosine similarity using their surrounding context vectors. We performed this experiment for German and Swedish and report average accuracies with and without this addition in Table 7. For monolingual corpora, we used the German and Swedish side of the Europarl corpus (Koehn, 2005) consisting of approximately 15 million and 21 million words respectively. We were able to project context vectors for an average of 4224.5 words in the two languages among all the possible compound words detected in Section 6.4. The poor Eurpoarl coverage could be due to the fact that compound words are generally technical words with low Europarl corpus frequency, especially in parliamentary proceedings. We believe that the small performance gains here are due to these limitations of the monolingual corpora.

Method	**Top1**$_{avg}$	**Top10**$_{avg}$
Original ranking	0.196	0.388
Comb. with Context Sim	0.201	0.391

Table 7: Average performance on German and Swedish with and without using context vector similarity from monolingual corpora.

6.6 Using phrase-tables if a parallel corpus is available

All previous results in this paper have been for translation lexicon discovery *without* the need for parallel bilingual text (bitext), which is often in limited supply for lower-resource languages. However, it is useful to assess how this translation lexicon dis-

covery work compares with traditional bitext-based lexicon induction (and how well the approaches can be combined). For this purpose, we used phrase tables learned by the standard statistical MT Toolkit Moses (Koehn et al., 2007). We tested the phrase-table accuracy on two languages, one for which we had a lot of parallel data available (German-English Europarl corpus with approx. 15 million words) and one for which we had relatively little parallel data (Czech-English news-commentary corpus with approx. 1 million words). This was done to see how the amount of parallel data available affects the accuracy and coverage of compound translation. Table 8 shows the performance for this experiment. For German, we see a significant improvement in accuracy and for Czech a small improvement in Top1 but a decline in Top10 accuracy. Note that these accuracies are still quite low as compared to general performance of phrase tables in an end-to-end MT system because we are measuring exact-match accuracy on a generally more challenging and often-lower-frequency lexicon subset. The third row in Table 8 for each of the languages shows that if one had a parallel corpus available, its n-best list can be combined with the n-best list of Bilingual Dictionaries algorithm to provide much higher consensus accuracy gains using weighted voting.

Method	# of words translated	Top1 Acc.	Top10 Acc.
German			
BiDict	13372	0.200	0.391
Parallel Corpus SMT	3281	0.423	0.576
Parallel + BiDict	3281	0.452	0.579
Czech			
$\text{BiDict}_{thresh=1}$	3455	0.276	0.514
Parallel Corpus SMT	309	0.285	0.404
Parallel + BiDict	309	0.359	0.599

Table 8: Performance of this paper's BiDict approach compared with and augmented with traditional statistical MT learning from bitext.

7 Quantifying the Role of Cross-languages

7.1 Coverage/Accuracy Trade off

The number of languages offering a translation hypothesis for a given literal English gloss is a useful parameter for measuring confidence in the algorithm's selection. The more distinct languages exhibiting a translation for the gloss, the higher likelihood that the majority translation will be correct rather than noise. Varying this parameter yields the coverage/accuracy trade off as shown in Figure 3.

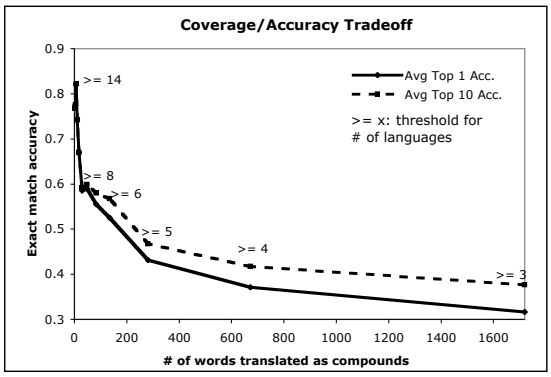


Figure 3: Coverage/Accuracy trade off curve by incrementing the minimum number of languages exhibiting a candidate translation for the source-word's literal English gloss. Accuracy here is the Top1 accuracy averaged over all 10 test languages.

7.2 Varying size of bilingual dictionaries

Figure 4 illustrates how the size of the bilingual dictionaries used for providing cross-language evidence affects translation performance. In order to take both coverage and accuracy into account, performance measure used was the F-score which is a harmonic average of Precision (the accuracy on the subset of words that could be translated) and Psuedo-recall (which is the correctly translated fraction out of total words that could be translated using 100% of the dictionary size). We can see in Figure 4 that increasing the percentage of dictionary size[7] always helps without plateauing, suggesting substantial extrapolation potential from large dictionaries.

7.3 Greedy vs Random Selection of Utilized Languages

A natural question for our compound translation algorithm is how does the choice of additional languages affect performance. We report two experiments on this question. A simple experiment is to use bilingual dictionaries of randomly selected languages and test the performance of K-randomly selected languages[8], incrementing K until it is the full set of 50 languages. The dashed lines in Figures 5

[7]Each run of choosing a percentage of dictionary size was averaged over 10 runs

[8]Each run of randomly selecting K languages was averaged over 10 runs.

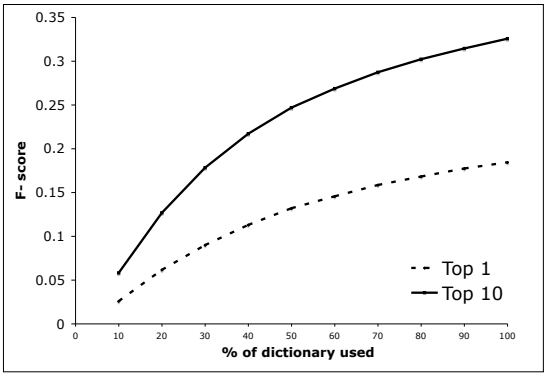


Figure 4: F-measure performance given varying sizes of the bilingual dictionaries used for cross-language evidence (as a percentage of words randomly utilized from each dictionary).

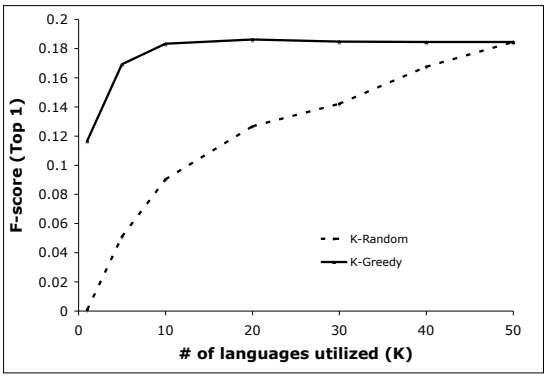


Figure 5: Top-1 match F-score performance utilizing K languages for cross-language evidence, for both a random K languages and greedy selection of the most effective K languages (typically the closest or largest dictionaries)

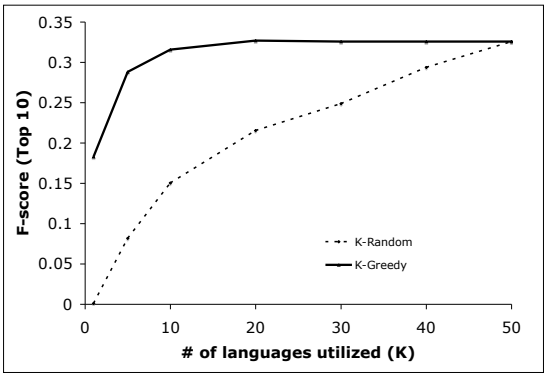


Figure 6: The performance relationship detailed in Figure 5 caption for Top-10 match F-score.

and 6 show this trend. The performance is measured by F-score as in section 7.1, where Pseudo-Recall here is the fraction of correct candidates out of the total candidates that could be translated had we used bilingual dictionaries of all the languages. We can see that adding random bilingual dictionaries helps improve the performance in a close to linear fashion. Furthermore, we observe that certain contributing languages are much more effective than others (e.g. Arabic/Farsi vs. Arabic/Czech). We use a greedy heuristic for ranking an additional cross-language, that is the number of test words for which the correct English translation can be provided by the bilingual dictionary of the respective cross-language. Figures 5 and 6 show that greedy selection of the most effective K utilized languages using this heuristic substantially accelerates performance. In fact, beyond the best 10 languages, performance plateaus and actually decreases slightly, indicating that increased noise is outweighing increased coverage.

Albanian			**Arabic**		
Russian	0.067	0.116	Farsi	0.051	0.090
+Spanish	0.100	0.169	+Spanish	0.059	0.111
+Bulgarian	0.119	0.201	+French	0.077	0.138
Bulgarian			**Czech**		
Russian	0.186	0.294	Slovak	0.177	0.289
+Hungarian	0.190	0.319	+Russian	0.222	0.368
+Swedish	0.203	0.339	+Hungarian	0.235	0.407
Farsi			**German**		
Arabic	0.031	0.047	Dutch	0.130	0.228
+Dutch	0.038	0.070	+Swedish	0.191	0.316
+Spanish	0.044	0.079	+Hungarian	0.204	0.355
Hungarian			**Russian**		
Swedish	0.073	0.108	Bulgarian	0.185	0.250
+Dutch	0.103	0.158	+Hungarian	0.199	0.292
+German	0.117	0.182	+Swedish	0.216	0.319
Slovak			**Swedish**		
Czech	0.145	0.218	German	0.120	0.188
+Russian	0.168	0.280	+Hungarian	0.152	0.264
+Hungarian	0.176	0.300	+Dutch	0.182	0.309

Table 9: Illustrating 3-best cross-languages obtained for each test language (shown in bold). Each row shows the effect of adding the respective cross-language to the set of languages in the rows above it and the corresponding F-scores (Top 1 and Top 10) achieved.

7.4 Languages found using Greedy selection

Table 9 shows the sets of the most effective three cross-languages per test language selected using the greedy heuristic explained in previous section. Unsurprisingly, related languages tend to help more than distant languages. For example, Dutch is most

effective for the test language German, and Slovak is most effective for Czech. We can also see interesting symmetries between related languages, for example: Farsi is the top language used for test language Arabic and vice-versa. Such symmetries can also be seen for other pairs of related languages such as (Czech, Slovak) and (Russian, Bulgarian). Thus, related languages are most helpful and they can be related in several ways such as etymologically, culturally and physically (such as Hungarian contact with the Germanic languages). The second point to note is that languages having large dictionaries also tend to be especially helpful, even when unrelated. This can be seen by the presence of Hungarian in top three cross-languages for most of the test languages. This is likely because Hungarian was one of the largest dictionaries and hence can provide good coverage for obtaining translation candidates of rarer or technical compounds, which may have more language universal literal glosses.

8 Conclusion

This paper has shown that successful translation of compounds can be achieved without the need for bilingual training text, by modeling the mapping of literal component-word glosses (e.g. "iron-path") into fluent English (e.g. "railway") across multiple languages. An interesting property of using such cross-language evidence is that one does need to restrict the candidate translations to compositional (or "glossy") translations, as our model allows the successful generation of more fluent non-compositional translations. We further show improved performance by adding component-sequence and learned-morphology models along with context similarity from monolingual text and optional combination with traditional bilingual-text-based translation discovery. These models show consistent performance gains across 10 diverse test languages.

9 Acknowledgments

We thank Chris Callison-Burch for providing access to phrase tables and giving valuable comments on this work as well as suggesting useful additional experiments. We also thank Markus Dreyer for helping with German examples and David Smith for giving valuable comments on initial version of the paper.

References

T. Baldwin and T. Tanaka. 2004. Translation by Machine of Complex Nominals: Getting it Right. *Proceedings of the ACL-2004 Workshop on Multiword Expressions*, pages 24–31.

R.D. Brown. 2002. Corpus-driven splitting of compound words. *Proceedings of TMI-2002*.

Y. Cao and H. Li. 2002. Base Noun Phrase translation using web data and the EM algorithm. *Proceedings of COLING-Volume 1*, pages 1–7.

G. Grefenstette. 1999. The World Wide Web as a Resource for Example-Based Machine Translation Tasks. *In ASLIB'99 Translating and the Computer 21*.

P. Koehn and K. Knight. 2003. Empirical methods for compound splitting. *Proceedings of the EACL-Volume 1*, pages 187–193.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. *Proceedings of ACL, companion volume*, pages 177–180.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *MT Summit X*.

J.N. Levi. 1978. *The Syntax and Semantics of Complex Nominals*.

R. Navigli, P. Velardi, and A. Gangemi. 2003. Ontology learning and its application to automated terminology translation. *Intelligent Systems, IEEE*, 18(1):22–31.

U. Rackow, I. Dagan, and U. Schwall. 1992. Automatic translation of noun compounds. *Proceedings of COLING-Volume 4*, pages 1249–1253.

R. Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. *Proceedings of ACL*, pages 519–526.

C. Schafer and D. Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. *Proceedings of COLING*, pages 1–7.

C. Schafer and D. Yarowsky. 2004. Exploiting aggregate properties of bilingual dictionaries for distinguishing senses of English words and inducing English sense clusters. *Proceedings of ACL-2004*, pages 118–121.

T. Tanaka and T. Baldwin. 2003. Noun-Noun Compound Machine Translation: A Feasibility Study on Shallow Processing. *Proceedings of the ACL-2003 Workshop on Multiword Expressions*, pages 17–24.

J. Zhang, J. Gao, and M. Zhou. 2000. Extraction of Chinese compound words: an experimental study on a very large corpus. *Proceedings of the Second Workshop on Chinese Language Processing*, pages 132–139.

Answering Definition Questions via Temporally-Anchored Text Snippets

Marius Paşca
Google Inc.
1600 Amphitheatre Parkway
Mountain View, California 94043
mars@google.com

Abstract

A lightweight extraction method derives text snippets associated to dates from the Web. The snippets are organized dynamically into answers to definition questions. Experiments on standard test question sets show that temporally-anchored text snippets allow for efficiently answering definition questions at accuracy levels comparable to the best systems, without any need for complex lexical resources, or specialized processing modules dedicated to finding definitions.

1 Introduction

In the field of automated question answering (QA), a variety of information sources and multiple extraction techniques can all contribute to producing relevant answers in response to natural-language questions submitted by users. Yet the nature of the information source which is mined for answers, together with the scope of the questions, have the most significant impact on the overall architecture of a QA system. When compared to the average queries submitted in a decentralized information seeking environment such as Web search, fact-seeking questions tend to specify better the nature of the information being sought by the user, whether it is the name of the longest river in some country, or the name of the general who defeated the Spanish Armada. In order to understand the structure and the linguistic clues encoded in natural-language questions, many QA systems employ sophisticated techniques, thus deriving useful information such as terms, relations among terms, the type of the expected answers (e.g., cities vs. countries vs. presidential candidates), and other semantic constraints (e.g., the elections from 1978 rather than any other year).

One class of questions whose characteristics place them closer to exploratory queries, rather than standard fact-seeking questions, are definition questions. Seeking information about an entity or a concept, questions such as *"Who is Caetano Veloso?"* offer little guidance as to what particular techniques could be used in order to return relevant information from a large text collection. In fact, the same user may choose to submit a definition question or a simpler exploratory query (*Caetano Veloso*), and still look for text snippets capturing relevant properties of the question concept. Various studies (Chen et al., 2006; Han et al., 2006) illustrate the challenges introduced by definition questions. As such questions have a less irregular form than other open-domain questions, recognizing their type is relatively easier (Hildebrandt et al., 2004). Conversely, the identification of relevant documents and the extraction of answers to definition questions are more laborious, and the impact on the architecture of QA systems is quite significant. Indeed, separate, dedicated modules, or even end-to-end systems are specifically built for answering definition questions (Klavans and Mureşan, 2001; Hildebrandt et al., 2004; Greenwood and Saggion, 2004). The importance of definition questions among other question categories is confirmed by their inclusion among the evaluation queries from the QA track of TREC evaluations (Voorhees and Tice, 2000).

This paper investigates the impact of temporally-

anchored text snippets derived from the Web, in answering definition questions and, more generally, exploratory queries. Section 2 describes a lightweight mechanism for extracting text snippets and associated dates from sentences in Web documents. Section 3 assesses the coverage of the extracted snippets. As shown in Section 4, relevant events, in which the question concept was involved, can be captured by matching the queries on the text snippets, and organizing the snippets around the associated dates. Section 5 describes discusses the role of the extracted text snippets in answering two sets of definition questions.

2 Temporally Anchored Text Snippets

All experiments rely on the unstructured text in approximately one billion documents in English from a 2003 Web repository snapshot of the Google search engine. Pre-processing of the documents consists in HTML tag removal, simplified sentence boundary detection, tokenization and part-of-speech tagging with the TnT tagger (Brants, 2000). No other tools or lexical resources are employed.

A sequence of sentence tokens represents a potential date if it consists of: single year (four-digit numbers, e.g., *1929*); or simple decade (e.g., *1930s*); or month name and year (e.g., *January 1929*); or month name, day number and year (e.g., *January 15, 1929*). Dates occurring in text in any other format are ignored. To avoid spurious matches, such as *1929 people*, potential dates are discarded if they are immediately followed by a noun or noun modifier, or immediately preceded by a noun.

To convert document sentences into a few text snippets associated with dates, the overall structure of sentences is roughly approximated. Deep text analysis may be desirable but simply not feasible on the Web. As a lightweight alternative, the proposed extraction method approximates the occurrence and boundaries of text snippets through the following set of lexico-syntactic patterns:
(P_1): $\langle Date$ [,|-|(|nil] [when] $Snippet$ [,|-|)|.]$\rangle$
(P_2): $\langle$[StartSent] [In|On] $Date$ [,|-|(|nil] $Snippet$ [,|-|)|.]$\rangle$
(P_3): $\langle$[StartSent] $Snippet$ [in|on] $Date$ [EndSent]$\rangle$
(P_4): $\langle$[Verb] [OptionalAdverb] [in|on] $Date\rangle$

The first extraction pattern, P_1, targets sentences with adverbial relative clauses introduced by wh-adverbs and preceded by a date, e.g.:

"By [$_{Date}$ 1910], when [$_{Snippet}$ Korea was annexed to Japan], the Korean population in America had grown to 5,008".

Comparatively, P_2 and P_3 match sentences that start or end in a simple adverbial phrase containing a date. In the case of P_4, the occurrence of relevant dates within sentences is approximated by verbs followed by a simple adverbial phrase containing a date. P_4 marks the entire sentence as a potential nugget because it lacks the punctuation clues in the other three patterns.

The patterns must satisfy additional constraints in order to match a sentence. These constraints constitute heuristics to avoid, rather than solve, complex linguistic phenomena. Thus, a nugget is always discarded if it does not contain a verb, or contains any pronoun. Furthermore, the snippets in P_2 and P_3 must start with, and the nugget in P_4 must contain a noun phrase, which in turn is approximated by the occurrence of a noun, adjective or determiner. The combination of patterns and constraints is by no means definitive or error-free. It is a practical solution to achieve graceful degradation on large amounts of data, reduce the extraction errors, and improve the usefulness of the extracted snippets. As such, it emphasizes robustness at Web scale, without taking advantage of existing specification languages for representing events and temporal expressions occurring in text (Pustejovsky et al., 2003), and forgoing the potential benefits of more complex methods that extract temporal relations from relatively clean text collections (Mani et al., 2006).

3 Coverage of Text Snippets

A concept such as a particular actor, country or organization usually occurs within more than one of the extracted text snippets. In fact, the set of text snippets containing the concept, together with the associated dates, often represents an extract-based, simple temporal summary of the events in which the concept has been involved. Starting from this observation, a task-based evaluation of the coverage of the extracted text snippets consists in verifying to what extent they capture the condensed history of several countries. Since any country must have been involved in some historical timeline of events, a reference timeline is readily available in an exter-

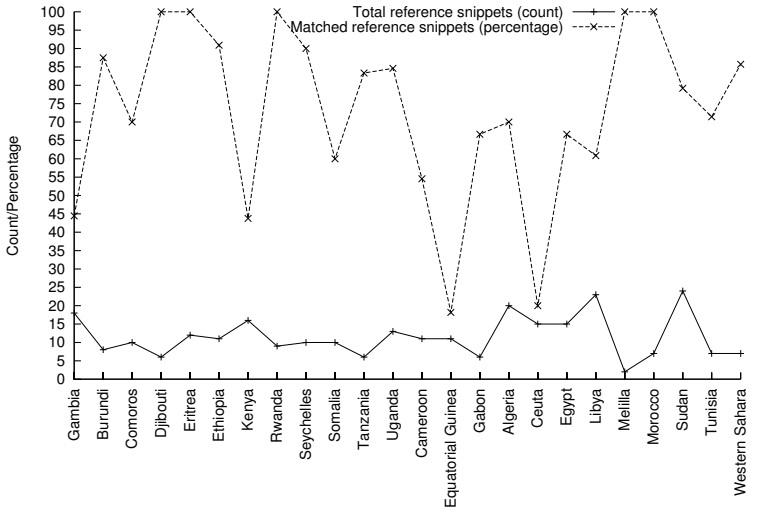


Figure 1: Percentage of reference snippets with corresponding extracted snippets

nal resource, e.g., encyclopedia, as an excerpt covering a condensed history of the country. The reference timeline is compared against the text snippets containing a country such as *Ethiopia*. To this effect, the text snippets containing a given country as a phrase are retained, ordered in increasing order of their associated dates, and evaluated against the reference timeline.

Both the test set of countries and the gold standard are collected from Wikipedia (Remy, 2002). The test set comprises countries from Africa. Since African countries have fewer extracted snippets than other countries, the evaluation results provide more useful, lower bounds rather than average or best-case. Due to limited human resources available for this evaluation, the test countries are a subset of the African countries in Wikipedia, selected in the order in which they are listed on the site. They cover all Eastern, Central and Northern Africa. The *Central African Republic*, the *Republic of the Congo*, and *Sao Tome and Principe* are discarded and *Gambia* added, leading to a test set of 24 country names. The source of the reference timelines is the condensed history article that is part of the main description page of each country in Wikipedia.

The evaluation procedure is concerned only with recall, but is still highly subjective. It requires the manual division of the reference text into dated events. In addition, the assessor must decide which details surrounding an event are significant, and must be matched into the extracted snippets in order to get any credit. The actual evaluation consists in matching each dated event from the reference timeline into the extracted timeline. During matching, the extracted snippets are analyzed by hand to decide which snippets, if any, capture the reference event, significant details around it, and the time stamp.

On average, 1173 text snippets are returned per country name, with a median of 733 snippets. Figure 1 summarizes the comparison of reference snippets and extracted snippets. The continuous line corresponds to the total number of reference snippets that were manually identified in the reference timeline; *Melilla* has the smallest such number (2), whereas *Sudan* has the largest (24). The dotted line in Figure 1 represents the percentage of reference snippets that have at least one match into the extracted snippets, thus evaluating recall. An average of 72% of the reference snippets have such matches. For 5 queries, there are matches for all reference snippets. The worst case occurs for *Equatorial Guinea*, for which only two out of the 11 reference snippets can be matched. Based on the results, we conclude that the text snippets and the associated dates provide a good coverage in the case of information about countries. The snippets can be retrieved as answers to questions asking about dates (*When*, *What year*) as described in (Paşca, 2007), or as answers to definition questions as discussed below.

4 Answering Definition Questions

Input definition questions are uniformly handled as Boolean queries, after the removal of stop words as well as question-specific terms (*Who* etc.). Thus, questions such as *"Who is Caetano Veloso?"* and *"Who won the Nobel Peace Prize?"* are consistently converted into conjunctive queries corresponding to *Caetano Veloso* and *won Nobel Peace Prize* respectively. The score assigned to a matching text snippet is higher, if the snippet occurs in a larger number of documents. Similarly, the score is higher if the snippet contains fewer non-stop terms in addition to the question term matches, or the average distance in the snippet between pairs of query term matches is lower. A side effect of the latter heuristic is to boost the snippets in which the query terms occur as a phrase, rather than as scattered term matches.

When they are associated to a common date, retrieved snippets transfer their relevance score onto the date, in the form of the sum of the individual snippet scores. The dates are ranked in decreasing order of their relevance scores, and those with the highest scores are returned as responses to the question, together with the top associated snippets. Within a set of text snippets associated to a date, the snippets are also ranked relatively to one another, such that each returned date is accompanied by its top supporting snippets. The ranking within a set of snippets associated to a date is a two-pass procedure. First, the snippets are scanned to count the number of occurrences of non-stop unigrams within the entire set. Second, a snippet is weighted with respect to others based on how many of the unigrams it contains, and the individual scores of those unigrams.

In the output, the snippets act as useful, implicit text-based justifications of why the dates may be relevant or not. As such, they implement a practical method of fusing together bits (snippets) of information collected from unrelated documents. In some cases, the snippets show why a returned result (date) is relevant. For example, *1990* is relevant to the query *Germany unified* because *"East and West Germany were unified"* according to the top snippet. In other cases, the text snippets quickly reveal why the result is related to the query even though it may not match the original user's intent. For instance, a user may ask the question *"When was the Taj Mahal built?"* with the well-known monument in mind, in which case the irrelevance of the date *1903* is self-explanatory based on one of its supporting snippets, *"the lavish Taj Mahal Hotel was built"*.

5 Evaluation

The answers returned by the system are ranked in decreasing order of their scores. By convention, an answer to a definition question comprises a returned date, plus the top matching text snippets that provide support for that date. Ideally, a snippet should only contain the desired answer and nothing else. In practice, a snippet is deemed correct if it contains the ideal answer, although it may contain some other extraneous information.

5.1 Objective Evaluation

A thorough evaluation of answers to definition questions would be complex, prone to subjective assessments, and would involve significant human labor (Voorhees, 2003). Therefore, the quality of the text snippets in the context of definition questions is tested on a set, DefQa1, containing the 23 *"Who is/was [ProperName]?"* questions from the TREC QA track from 1999 through 2002. In this case, each returned answer consists of a date and the first supporting text snippet.

Table 1 contains a sample of the test questions. The right column shows actual text snippets retrieved for the definition questions, together with the associated date and the rank of that date within the output. In an objective evaluation strictly based on the answer keys of the gold standard, the MRR score over the DefQa1 set is 0.596. The score is quite high, given that the answer keys prefer the genus of the question concept, rather than other types of information. For instance, the answer keys for the TREC questions Q222:*"Who is Anubis?"* and Q253:*"Who is William Wordsworth?"* mark *poet* and *"Egyptian god"* as correct answers respectively, thus emphasizing the genus of the question concepts *Anubis* and *William Wordsworth*. This explains the strong reliance in previous work on hand-written patterns and dictionary-based techniques for detecting text fragments encoding the genus and differentia of the question concept (Lin, 2002; Xu et al., 2004).

Question	(Rank) Relevant Date: Associated Fact
Q218: Who was Whitcomb Judson?	(1) 1893: First patented in 1893 by Whitcomb Judson, the Clasp Locker was notoriously unreliable and expensive
	(2) 1891: the zipper was invented by Whitcomb Judson
Q239: Who is Barbara Jordan?	(1) February 21 1936: Barbara Jordan was born in Houston, Texas
	(2) January 17 1996: Barbara Jordan died in Austin, Texas, at the age of 59
	(4) 1973: Barbara Jordan was diagnosed with multiple sclerosis and was confined to a wheelchair
	(5) 1976: Barbara Jordan became the first African-American Woman to deliver a keynote address at a political convention
	(7) 1966: Barbara Jordan became the first black representative since 1883 to win an election to the Texas legislature
	(8) 1972: Barbara Jordan was elected to the US Congress
Q253: Who is William Wordsworth?	(1) 1770: William Wordsworth was born in 1770 in the town of Cockermouth, England
	(2) April 7 1770: William Wordsworth was born
	(4) 1798: Romanticism officially began, when William Wordsworth and Samuel Taylor Coleridge anonymously published Lyrical Ballads
	(5) 1802: William Wordsworth married Mary Hutchinson at Brompton church
	(7) 1795: Coleridge met the poet William Wordsworth
	(8) April 23 1850: William Wordsworth died
	(11) 1843: William Wordsworth (1770-1850) was made Poet Laureate of Britain
Q346: Who is Langston Hughes?	(1) 1902: Langston Hughes was born in Joplin, Missouri
	(2) May 22 1967: Langston Hughes died of cancer
	(5) 1994: The Collected Poems of Langston Hughes was published
Q351: Who is Charles Lindbergh?	(1) 1927: aviation hero Charles Lindbergh was honored with a ticker-tape parade in New York City
	(2) 1932: Charles Lindbergh's infant son was kidnapped and murdered
	(3) February 4 1902: Charles Lindbergh was born in Detroit
	(5) August 26 1974: Charles Lindbergh died
	(7) May 21 1927: Charles Lindbergh landed in Paris
	(8) May 20 1927: Charles Lindbergh took off from Long Island
	(9) May 1927: an airmail pilot named Charles Lindbergh made the first solo flight across the Atlantic Ocean
Q419: Who was Jane Goodall?	(1) 1977: Goodall founded the Jane Goodall Institute for Wildlife Research
	(2) April 3 1934: Jane Goodall was born in London, England
	(3) 1960: Dr Jane Goodall began studying chimpanzees in east Africa
	(8) 1985: Jane Goodall 's twenty-five years of anthropological and conservation research was published

Table 1: Temporally-anchored text snippets returned as answers to definition questions

5.2 Subjective Evaluation

Beyond the snippets that happen to contain the genus of the question concept, the output constitutes supplemental results to what other definition QA systems may offer. The intuition is that prominent facts associated with the question concept provide useful, if not direct answers to the corresponding definition question, with the twist of presenting them together with the associated date. For instance, the first answer to Q239:*"Who is Barbara Jordan?"* reveals her date of birth and is associated with the first retrieved date, *February 21 1936*. In the objective evaluation, this answer is marked as incorrect. However, some users may find this snippet useful, although they may still prefer the seventh or eighth text snippets from Table 1 as primary answers, as they mention *Barbara Jordan*'s election to a state legislature in *1966*, and to the Congress in *1972*. As an alternative evaluation, the top five matching snippets for each of the top ten dates are inspected manually, and answers such as the birth year of a person are subjectively marked as correct. Overall, 59.1% of the snippets returned for the DefQa1 questions are deemed correct, which shows that the answers capture useful properties of the question concepts.

5.3 Alternative Objective Evaluation

A separate objective evaluation was conducted on a set, DefQa2, containing the 24 definition questions asking for information about various people, from the TREC QA track from 2004. Although correctness assessments are still subjective, they benefit from a more rigorous evaluation procedure. For each question, the gold standard consists of sets of responses classified according to their importance into two classes, namely *vital* nuggets, containing

information that the assessors feel must be returned for the overall output to be good, and *non-vital*, containing information that is acceptable in the output but not necessary.

Following the official 2004 evaluation procedure (Voorhees, 2004), a returned text snippet is considered vital, non-vital, or incorrect based on whether it conceptually matches a vital, non-vital answer, or none of the answers specified in the gold standard for that question. The overall recall is the average of individual recall values per question, which are computed as the number of returned vital answers, divided by the number of vital answers from the gold standard for a given question. In this case, a returned answer is formed by a date and its top three associated text snippets. If a vital answer from the gold standard matches any of the three snippets of a returned answer, then the returned answer is vital.

The overall recall value over DefQa2 is 0.46. The corresponding F-measure, which gives three times more importance to recall than to precision as specified in the official evaluation procedure, is 0.39. The score measures favorably against the top three F-measure scores of 0.46, 0.40, and 0.37 reported in the official 2004 evaluation (Voorhees, 2004). The two better scores were obtained by systems that rely extensively on human-generated knowledge from resources such as WordNet (Zhang et al., 2005) and specific Web glossaries (Cui et al., 2007). In comparison, the text snippets retrieved in this paper provide relevant answers to definition questions with the added benefit of providing a temporal anchor for each answer, and without using any complex linguistic resources and tools.

The scores per question vary widely, with the retrieved snippets containing none of the vital answers for six questions, all vital answers for other six, and some fraction of the vital answers for the remaining questions. For example, one of the retrieved text snippets is *"US Air Force Colonel Eileen Marie Collins was the first woman to command a space shuttle mission"*. The snippet is classified as vital for the question about *Eileen Marie Collins*, since it conceptually matches a vital answer from the gold standard, namely *"first woman space shuttle commander"*. Again, even though the standard evaluation does not require a temporal anchor for an answer to be correct, we feel that the dates associated to the retrieved snippets provide very useful, additional, condensed information. In the case of *Eileen Marie Collins*, the above-mentioned vital answer is accompanied by the date *1999*, when the mission took place.

6 Related Work

Previous approaches to answering definition questions from large text collections can be classified according to the kind of techniques for the extraction of answers. A significant body of work is oriented towards mining descriptive phrases or sentences, as opposed to other types of semantic information, for the given question concepts. To this effect, the use of hand-written lexico-syntactic patterns and regular expressions, targeting the genus and possibly the differentia of the question concept, is widespread, whether employed for mining definitions in English (Liu et al., 2003; Hildebrandt et al., 2004) or other languages such as Japanese (Fujii and Ishikawa, 2004), from local text collections (Xu et al., 2004) or from the Web (Blair-Goldensohn et al., 2004; Androutsopoulos and Galanis, 2005). Comparatively, the small set of patterns used here targets text snippets that are temporally-anchored. Therefore the text snippets provide answers to definition answers without actually employing any specialized module for seeking specific information such as the genus of the question concept.

Several studies propose unsupervised extraction methods as an alternative to using hand-written patterns for definition questions (Androutsopoulos and Galanis, 2005; Cui et al., 2007). Previous work often relies on external resources as an important or even essential guide towards the desired output. Such resources include WordNet (Prager et al., 2001) for finding the genus of the question concept; large dictionaries such as Merriam Webster, for ready-to-use definitions (Xu et al., 2004; Hildebrandt et al., 2004); and encyclopedias, for collecting words that are likely to occur in potential definitions (Fujii and Ishikawa, 2004; Xu et al., 2004). In comparison, the experiments reported in this paper do not require any external lexical resource.

7 Conclusion

Without specifically targeting definitions, temporally-anchored text snippets extracted from the Web provide very useful answers to definition questions, as measured on standard test question sets. Since the snippets tend to capture important events involving the question concepts, rather than phrases that describe the question concept, they can be employed as either standalone answers, or supplemental results in conjunction with answers extracted with other techniques.

References

I. Androutsopoulos and D. Galanis. 2005. A practically unsupervised learning method to identify single-snippet answers to definition questions on the Web. In *Proceedings of the Human Language Technology Conference (HLT-EMNLP-05)*, pages 323–330, Vancouver, Canada.

S. Blair-Goldensohn, K. McKeown, and A. Schlaikjer, 2004. *New Directions in Question Answering*, chapter Answering Definitional Questions: a Hybrid Approach, pages 47–58. MIT Press, Cambridge, Massachusetts.

T. Brants. 2000. TnT - a statistical part of speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, pages 224–231, Seattle, Washington.

Y. Chen, M. Zhou, and S. Wang. 2006. Reranking answers for definitional QA using language modeling. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 1081–1088, Sydney, Australia.

H. Cui, M. Kan, and T. Chua. 2007. Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems*, 25(2).

A. Fujii and T. Ishikawa. 2004. Summarizing encyclopedic term descriptions on the Web. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 645–651, Geneva, Switzerland.

M. Greenwood and H. Saggion. 2004. A pattern based approach to answering factoid, list and definition questions. In *Proceedings of the 7th Content-Based Multimedia Information Access Conference (RIAO-04)*, pages 232–243, Avignon, France.

K. Han, Y. Song, and H. Rim. 2006. Probabilistic model for definitional question answering. In *Proceedings of the 29th ACM Conference on Research and Development in Information Retrieval (SIGIR-06)*, pages 212–219, Seattle, Washington.

W. Hildebrandt, B. Katz, and J. Lin. 2004. Answering definition questions with multiple knowledge sources. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 49–56, Boston, Massachusetts.

J. Klavans and Smaranda Mureşan. 2001. Evaluation of Definder: A system to mine definitions from consumer-oriented medical text. In *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL-01)*, pages 201–203, Roanoke, Virginia.

C.Y. Lin. 2002. The effectiveness of dictionary and web-based answer reranking. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)*, pages 1–7, Taipei, Taiwan.

B. Liu, C. Chin, and H.T. Ng. 2003. Mining topic-specific concepts and definitions on the Web. In *Proceedings of the 12th International World Wide Web Conference (WWW-03)*, pages 251–260, Budapest, Hungary.

I. Mani, M. Verhagen, B. Wellner, C. Lee, and J. Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 753–760, Sydney, Australia.

M. Paşca. 2007. Lightweight Web-based fact repositories for textual question answering. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM-07)*, Lisboa, Portugal.

J. Prager, D. Radev, and K. Czuba. 2001. Answering what-is questions by virtual annotation. In *Proceedings of the 1st Human Language Technology Conference (HLT-01)*, pages 1–5, San Diego, California.

J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, A. Setzer, and G. Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the 5th International Workshop on Computational Semantics (IWCS-5)*, Tilburg, Netherlands.

M. Remy. 2002. Wikipedia: The free encyclopedia. *Online Information Review*, 26(6):434.

E.M. Voorhees and D.M. Tice. 2000. Building a question-answering test collection. In *Proceedings of the 23rd International Conference on Research and Development in Information Retrieval (SIGIR-00)*, pages 200–207, Athens, Greece.

E. Voorhees. 2003. Evaluating answers to definition questions. In *Proceedings of the 2003 Human Language Technology Conference (HLT-NAACL-03)*, pages 109–111, Edmonton, Canada.

E.M. Voorhees. 2004. Overview of the TREC-2004 Question Answering track. In *Proceedings of the 13th Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland. NIST.

J. Xu, R. Weischedel, and A. Licuanan. 2004. Evaluation of an extraction-based approach to answering definitional questions. In *Proceedings of the 27th ACM Conference on Research and Development in Information Retrieval (SIGIR-04)*, pages 418–424, Sheffield, United Kingdom.

Z. Zhang, Y. Zhou, X. Huang, and L. Wu. 2005. Answering definition questions using Web knowledge bases. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 498–506, Jeju Island, Korea.

Corpus-based Question Answering for *why*-Questions

Ryuichiro Higashinaka and **Hideki Isozaki**
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Kyoto 619-0237, Japan
{rh,isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes a corpus-based approach for answering *why*-questions. Conventional systems use hand-crafted patterns to extract and evaluate answer candidates. However, such hand-crafted patterns are likely to have low coverage of causal expressions, and it is also difficult to assign suitable weights to the patterns by hand. In our approach, causal expressions are automatically collected from corpora tagged with semantic relations. From the collected expressions, features are created to train an answer candidate ranker that maximizes the QA performance with regards to the corpus of why-questions and answers. NAZEQA, a Japanese why-QA system based on our approach, clearly outperforms a baseline that uses hand-crafted patterns with a Mean Reciprocal Rank (top-5) of 0.305, making it presumably the best-performing fully implemented why-QA system.

1 Introduction

Following the trend of non-factoid QA, we are seeing the emergence of work on why-QA; e.g., answering generic "why X?" questions (Verberne, 2006). However, since why-QA is an inherently difficult problem, there have only been a small number of fully implemented systems dedicated to solving it. Recent systems at NTCIR-6[1] Question Answering Challenge (QAC-4) can handle why-questions (Fukumoto et al., 2007). However, their performance is much lower (Mori et al., 2007) than that of factoid QA systems (Fukumoto et al., 2004; Voorhees and Dang, 2005).

We consider that this low performance is due to the great amount of hand-crafting involved in the systems. Currently, most of the systems rely on hand-crafted patterns to extract and evaluate answer candidates (Fukumoto et al., 2007). Such patterns include typical cue phrases and POS-tag sequences related to causality, such as "*because of*" and "*by reason of*." However, as noted in (Inui and Okumura, 2005), causes are expressed in various forms, and it is difficult to cover all such expressions by hand. Hand-crafting is also very costly. Some patterns may be more indicative of causes than others. Therefore, it may be useful to assign different weights to the patterns for better answer candidate extraction, but currently this must be done by hand (Mori et al., 2007). It is not clear whether the weights determined by hand are suitable.

In this paper, we propose a corpus-based approach for why-QA in order to reduce this hand-crafting effort. We automatically collect causal expressions from corpora to improve the coverage of causal expressions, and utilize a machine learning technique to train a ranker of answer candidates on the basis of features created from the expressions together with other possible features related to causality. The ranker is trained to maximize the QA performance with regards to a corpus of why-questions and answers, automatically tuning the weights of the features.

This paper is organized as follows: Section 2 describes previous work on why-QA, and Section 3 describes our approach. Section 4 describes the implementation of our approach, and Section 5 presents the evaluation results. Section 6 summarizes and mentions future work.

2 Previous Work

Although systems that can answer why-questions are emerging, they tend to have limitations in that they can answer questions only with causal verbs (Girju, 2003), in specific domains (Khoo et al.,

[1] http://research.nii.ac.jp/ntcir/ntcir-ws6/ws-en.html

2000), or questions covered by a specific knowledge base (Curtis et al., 2005). Recently, Verberne (2006; 2007a) has been intensively working on why-QA based on the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). However, her approach requires manually annotated corpora with RST relations.

When we look for fully implemented systems for generic "why X?" questions, we only find a small number of such systems. Since why-QA would be a challenging task when tackled straightforwardly, requiring common-sense knowledge and semantic interpretation of questions and answer candidates, current systems place higher priority on achievability and therefore use hand-crafted patterns and heuristics to extract causal expressions as answer candidates and use conventional sentence similarity metrics for answer candidate evaluation (Fukumoto, 2007; Mori et al., 2007). We argue, in this paper, that this hand-crafting is the cause of the current low performance levels. Recently, (Shima and Mitamura, 2007) applied a machine learning approach to why-QA, but they also rely on manually selected cue words to create their features.

Semantic Role Labeling (SRL) techniques can be used to automatically detect causal expressions. In the CoNLL-2005 shared task (SRL for English), the best system found causal adjuncts with a reasonable accuracy of 65% (Màrquez et al., 2005). However, when we analyzed the data, we found that more than half of the causal adjuncts contain explicit cues such as "*because*." Since causes are reported to be expressed by a wide variety of linguistic phenomena, not just explicit cues (Inui and Okumura, 2005), further verification is needed before SRL can be safely used for why-QA.

Why-questions are a subset of non-factoid questions. Since non-factoid questions are observed in many FAQ sites, such sites have been regarded as valuable resources for the development of non-factoid QA systems. Examples include Burke et al. (1997), who used FAQ corpora to analyze questions to achieve accurate question-type matching; Soricut and Brill (2006), who used them to train statistical models for answer evaluation and formulation; and Mizuno et al. (2007), who used them to train classifiers of question and answer-types. However, they do not focus on why-questions and do not use any causal knowledge, which is considered to be useful for explicit why-questions (Soricut and Brill, 2006).

3 Approach

In this paper, we propose a corpus-based approach for why-QA in order to reduce the hand-crafting effort that is currently necessary. We first automatically collect causal expressions from corpora and use them to create features to represent an answer candidate. The features are then used to train an answer candidate ranker that maximizes the QA performance with regards to a corpus of why-questions and answers. We also enumerate possible features that may be useful for why-QA to be incorporated in the training to improve the QA performance.

Following the systems at QAC-4 (Fukumoto, 2007) and the answer analysis in (Verberne, 2007b; Verberne et al., 2007), we consider the task of why-QA to be a sentence/paragraph extraction task. We also assume that a document retrieval module of a system returns top-N documents for a question on the basis of conventional IR-related metrics and all sentences/paragraphs extracted from them are regarded as answer candidates. Hence, the task becomes the ranking of given sentences/paragraphs.

For an answer candidate (a sentence or a paragraph) to be the correct answer, the candidate should (1) have an expression indicating a cause and (2) be similar to the question in content, and (3) some causal relation should be observed between the candidate and the question. For example, an answer candidate "X was arrested for fraud." is likely to be a correct answer to the question "Why was X arrested?" because "for fraud" expresses a cause, the question and the answer are both about the same event (X being arrested), and "fraud" and "arrest" indicate a causal relation between the question and the candidate. Condition (3) would be especially useful when the candidates do not have obvious cues or topically similar words/phrases to the question; it may be worthwhile to rely on some prior causal knowledge to select one over others. Although current working systems (Fukumoto, 2007; Mori et al., 2007) do not explicitly state these conditions, they can be regarded as using hand-crafted patterns for (1) and (3).[2] Lexical similarity metrics, such as cosine similarity and n-gram overlaps, are generally used for (2).

We represent each answer candidate with causal expression, content similarity, and causal relation

[2](3) is dealt with in a manner similar to the treatment of '`cause_of_death`' in (Smith et al., 2005).

features that encode how it complies with the three conditions. Here, the causal expression features are those based on the causal expressions we aim to collect automatically. For the other two types of features, we turn to the existing similarity metrics and dictionaries to derive features that would be useful for why-QA. To train a ranker, we create a corpus of why-questions and answers and adopt one of the machine learning algorithms for ranking. The following sections describe the three types of features, the corpus creation, and the ranker training. The actual instances of the features, the corpus, and the ranker will be presented in Section 4.

3.1 Causal Expression Features

With the increasing attention paid to SRL, we currently have a number of corpora, such as PropBank (Palmer, 2005) and FrameNet (Baker et al., 1998), that are tagged with semantic relations including a causal relation. Since text spans for such relations are annotated in the corpora, we can simply collect the spans marked by a causal relation as causal expressions. Since an answer candidate that has a matching expression for one of the collected causal expressions is likely to be expressing a cause as well, we can make the existence of each expression a feature. Although the collected causal expressions without any modification might be used to create features, for generality, it would be better to abstract them into syntactic patterns. From *m* causal expressions/patterns automatically extracted from corpora, we can create *m* binary features.

In addition, some why-QA systems may already possess some good hand-crafted patterns to detect causal expressions. Since there is no reason not to use them if we know they are useful for why-QA, we can create a feature indicating whether an answer candidate matches existing hand-crafted patterns.

3.2 Content Similarity Features

In general, if a question and an answer candidate share many words, it is likely that they are about the same content. From this assumption, we create a feature that encodes the lexical similarity of an answer candidate to the question. To calculate its value, existing sentence similarity metrics, such as cosine similarity or n-gram overlaps, can be used.

Even if a question and an answer candidate do not share the same words, they may still be about the same content. One such case is when they are about the same topic. To express this case as a feature, we can use the similarity of the question and the document in which the answer candidate is found. Since the documents from which we extract answer candidates typically have scores output by an IR engine that encode their relevance to the question, we can use this score or simply the rank of the retrieved document as a feature.

A question and an answer candidate may be semantically expressing the same content with different expressions. The simplest case is when synonyms are used to describe the same content; e.g., when "arrest" is used instead of "apprehend." For such cases, we can exploit existing thesauri. We can create a feature encoding whether synonyms of words in the question are found in the answer candidate. We could also use the value of semantic similarity and relatedness measures (Pedersen et al., 2004) or the existence of hypernym or hyponym relations as features.

3.3 Causal Relation Features

There are semantic lexicons where a semantic relation between concepts is indicated. For example, the EDR dictionary[3] shows whether a causal relation holds between two concepts; e.g., between "murder" and "arrest." Using such dictionaries, we can create pairs of expressions, one indicating a cause and the other its effect. If we find an expression for a cause in the answer candidate and that for an effect in the question, it is likely that they hold a causal relation. Therefore, we can create a feature encoding whether this is the case. In cases where such semantic lexicons are not available, they may be automatically constructed, although with noise, using causal mining techniques such as (Marcu and Echihabi, 2002; Girju, 2003; Chang and Choi, 2004).

3.4 Creating a QA Corpus

For ranker training, we need a corpus of why-questions and answers. Because we regard the task of why-QA as a ranking of given sentences/paragraphs, it is best to prepare the corpus in the same setting. Therefore, we use the following procedure to create the corpus: (a) create a question, (b) use an IR engine to retrieve documents for the question, (c) select among all sentences/paragraphs in the retrieved documents those that contain the answer to the question, and (d) store the question and a

[3] http://www2.nict.go.jp/r/r312/EDR/index.html

set of selected sentences/paragraphs with their document IDs as answers.

3.5 Training a Ranker

Having created the QA corpus, we can apply existing machine learning algorithms for ranking, such as RankBoost (Freund et al., 2003) or Ranking SVM (Joachims, 2002), so that the selected sentences/paragraphs are preferred to non-selected ones on the basis of their features. Good ranking would result in good Mean Reciprocal Rank (MRR), which is one of the most commonly used measures in QA.

4 Implementation

Using our approach, we implemented a Japanese why-QA system, **NAZEQA** ("Naze" means "why" in Japanese). The system was built as an extension to our factoid QA system, SAIQA (Isozaki, 2004; Isozaki, 2005), and works as follows:

1. The question is analyzed by a rule-based question analysis component to derive a question type; 'REASON' for a why-question.
2. The document retrieval engine extracts n-best documents from Mainichi newspaper articles (1998–2001) using DIDF (Isozaki, 2005), a variant of the IDF metric. We chose 20 as n. All sentences/paragraphs in the n documents are extracted as answer candidates. Whether to use sentences or paragraphs as answer candidates is configurable.
3. The feature extraction component produces, for each answer candidate, causal expression, content similarity, and causal relation features encoding how it satisfies conditions (1)–(3) described in Section 3.
4. The SVM ranker trained by a QA corpus ranks the answer candidates based on the features.
5. The top-N answer candidates are presented to the user as answers.

In the following sections, we describe the features (399 in all), the QA corpus, and the ranker.

4.1 Causal Expression Features

(F1–F394: AUTO-Causal Expression) We automatically extracted causal expressions from the EDR dictionary. The EDR dictionary is a suite of corpora and dictionaries and includes the EDR corpus, the EDR concept dictionary (hierarchy of word senses), and the EDR Japanese word dictionary (sense to word mappings). The EDR corpus is a collection of independent Japanese sentences taken from various sources, such as newspaper articles, magazines, and dictionary glosses. The corpus is annotated with semantic relations including a causal relation in a manner similar to PropBank and FrameNet corpora. We extracted regions marked by 'cause' tags and abstracted them by leaving only the functional words (auxiliary verbs and case, aspect, tense markers) and replacing others with wildcards '*.' For example, a causal expression "arrested for fraud" would be abstracted to "*-PASS for *." We used CaboCha[4] as a morphological analyzer. From 8,747 regions annotated with 'cause,' we obtained 394 causal expression patterns after filtering out those that occurred only once. Finally, we have 394 binary features representing the existence of each abstracted causal expression pattern.

(F395: MAN-Causal Expression) We emulate the manually created patterns described in (Fukumoto, 2007) and create a binary feature indicating whether an answer candidate is matched by the patterns.

4.2 Content Similarity Features

(F396: Question-Candidate Cosine Similarity) We use the cosine similarity between a question and an answer candidate using the word frequency vectors of the content words. We chose nouns, verbs, and adjectives as content words.

(F397: Question-Document Relevance) We use, as a feature, the inverse of the rank of the document where the answer candidate is found.

(F398: Synonym Pair) This is a binary feature that indicates whether a word and its synonym appear in an answer candidate and a question, respectively. We use the combination of the EDR concept dictionary and the EDR Japanese word dictionary as a thesaurus to collect synonym pairs. We have 133,486 synonym pairs.

4.3 Causal Relation Feature

(F399: Cause-Effect Pair) This is a binary feature that indicates whether a word representing a cause and a word corresponding to its effect appear in an answer candidate and a question, respectively. We used the EDR concept dictionary to find pairs of word senses holding a causal relation and

[4] http://chasen.org/~taku/software/cabocha/

Q13: Why are pandas on the verge of extinction? (000217262)

A:000217262,L2 Since pandas are not good at raising their offspring, the Panda Preservation Center in Sichuan Province is promoting artificial insemination as well as the training of mother pandas.

A:000217262,L3 A mother panda often gives birth to two cubs, but when there are two cubs, one is discarded, and young mothers sometimes crush their babies to death.

A:000406060,L6 However, because of the recent development in the midland, they are becoming extinct.

A:010219075,L122 The most common cause of the extinction for mammals, birds, and plants is degradation and destruction of habitat, followed by hunting and poaching for mammals and the impact of alien species for birds.

Figure 1: An excerpt from the WHYQA collection. The number in parentheses is the ID of the document used to come up with the question. The answers were headed by the document ID and the line number where the sentence is found in the document. (N.B. The above sentences were translated by the authors.)

expanded the senses to corresponding words using the EDR Japanese word dictionary to create cause-effect word pairs. We have 355,641 cause-effect word pairs.

4.4 WHYQA Collection

Since QAC-4 does not provide official answer sets and their questions include only a small number of why-questions, we created a corpus of why-questions and answers on our own.

An expert, who specializes in text analysis and is not one of authors, created questions from articles randomly extracted from Mainichi newspaper articles (1998–2001). Then, for each question, she created sentence-level answers by selecting the sentences that she considered to fully include the answer from a list of sentences from top-20 documents returned from the text retrieval engine with the question as input. Paragraph-level answers were automatically created from the sentence-level answers by selecting the paragraphs containing the answer sentences.

The analyst was instructed not to create questions by simply converting existing declarative sentences into interrogatives. It took approximately five months to create 1,000 question and answer sets (called the WHYQA collection). All questions are guaranteed to have answers. Figure 1 lists an example question and answer sentences in the collection.

4.5 Training a Ranker by Ranking SVM

Using the WHYQA collection, we trained ranking models using the ranking SVM (Joachims, 2002) (with a linear kernel) that minimizes the pairwise ranking error among the answer candidates. In the training data, the answers were labeled '+1' and non-answers '-1.' When using sentences as answers, there are 4,849 positive examples and 521,177 negative examples. In the case of paragraphs, there are 4,371 positive examples and 261,215 negative examples.

5 Evaluation

For evaluation, we compared the proposed system (**NAZEQA**) with two baselines. Baseline-1 (**COS**) simply uses, for answer candidate evaluation, the cosine similarity between an answer candidate and a question based on frequency vectors of their content words. The aim of having this baseline is to see how the system performs without any use of causal knowledge. Baseline-2 (**FK**) uses hand-crafted patterns described in (Fukumoto, 2007) to narrow down the answer candidates to those having explicit causal expressions, which are then ranked by the cosine similarity to the question. NAZEQA and the two baselines used the same document retrieval engine to obtain the top-20 documents and ranked the sentences or paragraphs in these documents.

5.1 Results

We made each system output the top-1, 5, 10, and 20 answer sentences and paragraphs for all 1,000 questions in the WHYQA collection. We used the MRR and coverage as the evaluation metrics. **Coverage** means the rate of questions that can be answered by the top-N answer candidates. Table 1 shows the MRRs and coverage for the baselines and NAZEQA. A 10-fold cross validation was used for the evaluation of NAZEQA.

We can see from the table that NAZEQA is better in all comparisons. A statistical test (a sign test that compares the number of times one system places the correct answer before the other) showed that NAZEQA is significantly better than FK for the top-5, 10, and 20 answers in the sentence and paragraph-levels ($p<0.01$). Although the sentence-level MRR for NAZEQA is rather low, the paragraph-level MRR for the top-5 answers is 0.305, which is reasonably high for a non-factoid QA system (Mizuno et al., 2007). The coverage is also

	MRR			Coverage		
top-N	COS	FK	**NZQ**	COS	FK	**NZQ**
Sentences as answer candidates:						
top-1	0.036	0.091+	0.113	3.6%	9.1%	11.3%
top-5	0.086	0.139+	**0.196***	19.1%	23.1%	**35.4%**
top-10	0.102	0.149+	**0.216***	31.3%	30.7%	**50.4%**
top-20	0.115	0.152	**0.227***	51.4%	35.5%	**66.6%**
Paragraphs as answer candidates:						
top-1	0.065	0.152+	0.186	6.5%	15.2%	18.6%
top-5	0.140	0.245+	**0.305***	29.2%	41.6%	**53.1%**
top-10	0.166	0.257+	**0.328***	48.8%	50.5%	**70.3%**
top-20	0.181	0.262+	**0.339***	70.7%	56.4%	**85.6%**

Table 1: Mean Reciprocal Rank (MRR) and coverage for the baselines (COS and FK) and the proposed NAZEQA (NZQ in the table) system for the entire WHYQA collection. The top-1, 5, 10, and 20 mean the numbers of topmost candidates used to calculate MRR and coverage. Asterisks indicate NAZEQA's statistical significance ($p<0.01$) over FK, and '+' FK's over COS.

Feature Set	Sent.	Para.
All features (**NAZEQA**)	0.181	0.287
w/o F1–F394 (AUTO-Causal Exp.)	**0.138***	**0.217***
w/o F395 (MAN-Causal Exp.)	0.179	0.286
w/o F396 (Q-Cand. Cosine Similarity)	**0.131***	**0.188***
w/o F397 (Doc.-Q Relevance)	0.161	0.275
w/o F398 (Synonym Pair)	0.180	0.282
w/o F399 (Cause-Effect Pair)	*0.184*	0.287

Table 2: Performance changes in MRR (top-5) when we exclude one of the feature sets. Asterisks indicate a statistically significant drop in performance from NAZEQA. In this experiment, we used a two-fold cross validation to reduce computational cost.

high for NAZEQA, making it possible to find answers within the top-10 sentences and top-5 paragraphs for more than 50% of the questions. Because there are no why-QA systems known to be better than NAZEQA in MRR and coverage and because NAZEQA clearly outperforms a competitive baseline (FK), we conclude that NAZEQA has one of the best performance levels for why-QA.

It is interesting to know how each of the feature sets (e.g., AUTO-Causal Expression Features) contributes to the QA performance. Table 2 shows how the performance in MRR (top-5) changes when one of the feature sets is excluded in the training. Although the drop in performance by removing the Question-Candidate Cosine Similarity feature is understandable, the performance also drops significantly from NAZEQA when we exclude AUTO-Causal Expression features, showing the effectiveness of our automatically collected causal patterns.

Rank	Feature Name	Weight
1	Question-Candidate Cosine Similarity	4.66
2	Exp.[*de* (by) * *wo* (-ACC) * *teshimai* (-PERF)]	1.86
3	Exp.[*no* (of) * *niyote wa* (according to)]	1.44
4	Exp.[*no* (of) * *na* (AUX) * *no* (of) * *de* (by)]	1.42
5	Exp.[*no* (of) * *ya* (or) * *niyotte* (by)]	1.35
6	Exp.[*no* (of) * *ya* (or) * *no* (of) * *de* (by)]	1.30
7	Exp.[*na* (AUX) * *niyotte* (by)]	1.23
8	Exp.[*koto niyotte* (by the fact that)]	1.22
9	Exp.[*to* (and) * *no* (of) * *niyotte* (by)]	1.20
10	Document-Question Relevance	0.89
	⋮	
27	Synonym Pair	0.40
102	MAN-Causal Expression	0.16
127	Cause-Effect Pair	0.15

Table 3: Weights of features learned by the ranking SVM. 'AUTO-Causal Expression' is denoted as 'Exp.' for lack of space. AUX means an auxiliary verb. The abstracted causal expression patterns are shown in square brackets with their English translations in parentheses.

The MAN-Causal Expression, Synonym Pair, and Cause-Effect Pair features, do not seem to contribute much to the performance. One of the reasons for the small contribution of the MAN-Causal Expression feature may be that the manual patterns used to create this feature overlap greatly with the automatically collected causal expression patterns, lowering the impact of the MAN-Causal Expression feature. The small contribution of the Synonym Pair feature is probably attributed to the way the answers were created in the creation of the WHYQA Collection. Since the answer candidates from which the expert chose the answers were those retrieved by a text retrieval engine that uses lexical similarity to retrieve relevant documents, it is possible that the answers that contain synonyms had already been filtered out in the beginning, making the Synonym Pair feature less effective. Without the Cause-Effect Pair feature, the performance does not change or even improves a little when sentences are used as answers. The reason for this may be that the syntactically well-formed sentences of the newspaper articles might have made causal cues and patterns more effective than prior causal knowledge. We need to investigate the difference between the manually created causal patterns and the automatically collected ones. We also need to investigate whether the Synonym Pair and Cause-Effect Pair features could be useful in other conditions; e.g., when answers are created in different ways. We also need to examine the quality of our synonym and cause-effect word pairs because

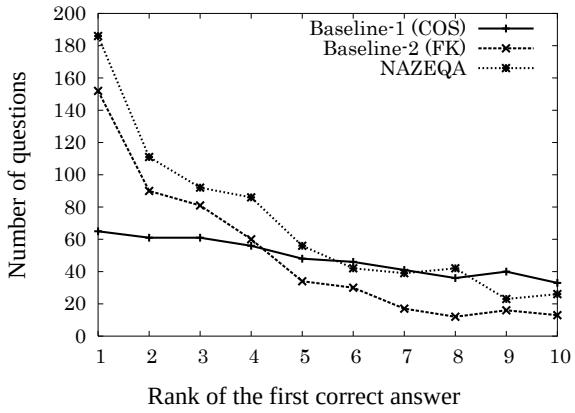


Figure 2: Distribution of the ranks of first correct answers. Paragraphs were used as answers. A 10-fold cross validation was used to evaluate NAZEQA.

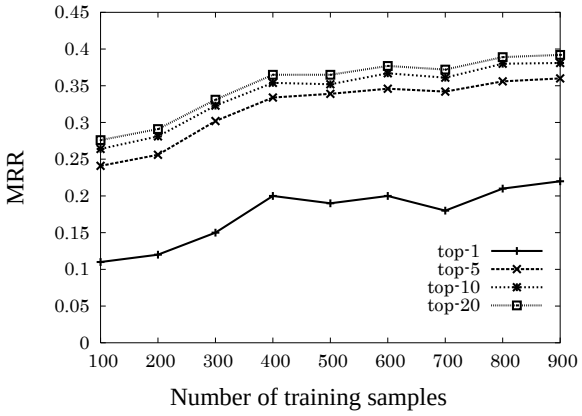


Figure 3: Learning curve: Performance changes when answering Q1–Q100 with different sizes of training samples. Paragraphs are used as answer candidates.

their quality itself may be to blame.

Furthermore, analyzing the trained ranking models allows us to calculate the weights given to the features (Hirao et al., 2002). Table 3 shows the weights of the top-10 features. We also include in the table the weights of the Synonym Pair, MAN-Causal Expression and Cause Effect Pair features so that the role of all three types of features in our approach can be shown. The analyzed model was the one trained with all 1,000 questions in the WHYQA collection with paragraphs as answers. Just as suggested by Table 2, the Question-Candidate Cosine Similarity feature plays a key role, followed by automatically collected causal expression features.

Figure 2 shows the distribution of the ranks of the first correct answers for all questions in the WHYQA collection for COS, FK, and NAZEQA. The distribution of COS is almost uniform, indicating that lexical similarity cannot be directly translated into causality. The figure also shows that NAZEQA consistently outperforms FK.

It may be useful to know how much training data is needed to train a ranker. We therefore fixed the test set to Q1–Q100 in the WHYQA collection and trained rankers with nine different sizes of training data (100–900) created from Q101–$\{$Q200 $\cdots$ Q1000$\}$. Figure 3 shows the learning curve. Naturally, the performance improves as we increase the data. However, the performance gains begin to decrease relatively early, possibly indicating the limitation of our approach. Since our approach heavily relies on surface patterns, the use of syntactic and semantic features may be necessary.

6 Summary and Future Work

This paper proposed corpus-based QA for why-questions. We automatically collected causal expressions from semantically tagged corpora and used them to create features to train an answer candidate ranker that maximizes the QA performance with regards to the corpus of why-questions and answers. The implemented system NAZEQA outperformed baselines with an MRR (top-5) of 0.305 and the coverage was also high, making NAZEQA presumably the best-performing system as a fully implemented why-QA system.

As future work, we are planning to investigate other features that may be useful for why-QA. We also need to examine how QA performance and the weights of the features differ when we use other sources for answer retrieval. In this work, we focused only on the '`cause`' relation in the EDR corpus to obtain causal expressions. However, there are other relations, such as '`purpose`,' that may also be related to causality (Verberne, 2006).

Although we believe our approach is language-independent, it would be worth verifying it by creating an English version of NAZEQA based on causal expressions that can be derived from PropBank and FrameNet. Finally, we are planning to make public some of the WHYQA collection at the authors' webpage so that various why-QA systems can be compared.

Acknowledgments

We thank Jun Suzuki, Kohji Dohsaka, Masaaki Nagata, and all members of the Knowledge Processing

Research Group for helpful discussions and comments. We also thank the anonymous reviewers for their valuable suggestions.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proc. COLING-ACL*, pages 86–90.

Robin Burke, Kristian Hammond, Vladimir Kulyukin, Steve Lytinen, Noriko Tomuro, and Scott Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the FAQFinder system. *AI Magazine*, 18(2):57–66.

Du-Seong Chang and Key-Sun Choi. 2004. Causal relation extraction using cue phrase and lexical pair probabilities. In *Proc. IJCNLP*, pages 61–70.

Jon Curtis, Gavin Matthews, and David Baxter. 2005. On the effective use of Cyc in a question answering system. In *Proc. IJCAI Workshop on Knowledge and Reasoning for Answering Questions*, pages 61–70.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Jun'ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. 2004. Question answering challenge for five ranked answers and list answers – overview of NTCIR4 QAC2 subtask 1 and 2 –. In *Proc. NTCIR*, pages 283–290.

Jun'ichi Fukumoto, Tsuneaki Kato, Fumito Masui, and Tsunenori Mori. 2007. An overview of the 4th question answering challenge (QAC-4) at NTCIR workshop 6. In *Proc. NTCIR*, pages 483–440.

Jun'ichi Fukumoto. 2007. Question answering system for non-factoid type questions and automatic evaluation based on BE method. In *Proc. NTCIR*, pages 441–447.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proc. ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 76–83.

Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. 2002. Extracting important sentences with support vector machines. In *Proc. 19th COLING*, pages 342–348.

Takashi Inui and Manabu Okumura. 2005. Investigating the characteristics of causal relations in Japanese text. In *Proc. ACL 2005 Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*.

Hideki Isozaki. 2004. NTT's question answering system for NTCIR QAC2. In *Proc. NTCIR*, pages 326–332.

Hideki Isozaki. 2005. An analysis of a high-performance Japanese question answering system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(3):263–279.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. KDD*, pages 133–142.

Christopher S. G. Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proc. 38th ACL*, pages 336–343.

W. Mann and S. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. In *Text*, volume 8, pages 243–281.

Daniel Marcu and Abdessamad Echihabi. 2002. In *Proc. 40th ACL*, pages 368–375.

Lluís Màrquez, Pere Comas, Jesús Giménez, and Neus Català. 2005. Semantic role labeling as sequential tagging. In *Proc. CoNLL*, pages 193–196.

Junta Mizuno, Tomoyosi Akiba, Atsushi Fujii, and Katunobu Itou. 2007. Non-factoid question answering experiments at NTCIR-6: Towards answer type detection for realworld questions. In *Proc. NTCIR*, pages 487–492.

Tatsunori Mori, Mitsuru Sato, Madoka Ishioroshi, Yugo Nishikawa, Shigenori Nakano, and Kei Kimura. 2007. A monolithic approach and a type-by-type approach for non-factoid question-answering – Yokohama National University at NTCIR-6 QAC –. In *Proc. NTCIR*, pages 469–476.

Martha Palmer. 2005. The proposition bank: An annotated corpus of semantic roles. *Comp. Ling.*, 31(1):71–106.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::Similarity - Measuring the Relatedness of Concepts. In *Proc. HLT-NAACL (Demonstration Papers)*, pages 38–41.

Hideki Shima and Teruko Mitamura. 2007. JAVELIN III: Answering non-factoid questions in Japanese. In *Proc. NTCIR*, pages 464–468.

Troy Smith, Thomas M. Repede, and Steven L. Lytinen. 2005. Determining the plausibility of answers to questions. In *Proc. AAAI Workshop on Inference for Textual Question Answering*, pages 52–58.

Radu Soricut and Eric Brill. 2006. Automatic question answering using the web: Beyond the factoid. *Journal of Information Retrieval*, 9:191–206.

Suzan Verberne, Lou Boves, Nelleke Oostdijk, and Peter-Arno Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proc. SIGIR (Posters and Demonstrations)*, pages 735–736.

Suzan Verberne. 2006. Developing an approach for why-question answering. In *Proc. 11th European Chapter of ACL*, pages 39–46.

Suzan Verberne. 2007a. Evaluating answer extraction for why-QA using RST-annotated Wikipedia texts. In *Proc. 12th ESSLLI Student Session*, pages 255–266.

Suzan Verberne. 2007b. Paragraph retrieval for why-question answering. In *Proc. Doctoral Consortium Workshop at SIGIR-2007*, page 922.

Ellen M. Voorhees and Hoa Trang Dang. 2005. Overview of the TREC 2005 question answering track. In *Proc. TREC*.

Cluster-Based Query Expansion for Statistical Question Answering

Lucian Vlad Lita ♠
Siemens Medical Solutions
lucian.lita@siemens.com

Jaime Carbonell
Carnegie Mellon University
jgc@cs.cmu.edu

Abstract

Document retrieval is a critical component of question answering (QA), yet little work has been done towards statistical modeling of queries and towards automatic generation of high quality query content for QA. This paper introduces a new, cluster-based query expansion method that learns queries known to be successful when applied to similar questions. We show that cluster-based expansion improves the retrieval performance of a statistical question answering system when used in addition to existing query expansion methods. This paper presents experiments with several feature selection methods used individually and in combination. We show that documents retrieved using the cluster-based approach are inherently different than documents retrieved using existing methods and provide a higher data diversity to answers extractors.

1 Introduction

Information retrieval has received sporadic examination in the context of question answering (QA). Over the past several years, research efforts have investigated retrieval quality in very controlled scenarios under the question answering task. At a first glance, document and passage retrieval is reasonable when considering the fact that its performance is often above 80% for this stage in the question answering process. However, most often, performance is measured in terms of the presence of at least one relevant document in the retrieved document set, regardless of relevant document density – where a document is relevant if it contains at least one correct answer. More specifically, the retrieval stage is considered successful even if there is a single document retrieved that mentions a correct answer, regardless of context. This performance measure is usually not realistic and revealing in question answering.

In typical scenarios, information extraction is not always able to identify correct answers in free text. When successfully found, correct answers are not always assigned sufficiently high confidence scores to ensure their high ranks in the final answer set. As a result, overall question answering scores are still suffering and considerable effort is being directed towards improving answer extraction and answer merging, yet little attention is being directed towards retrieval.

A closer look at retrieval in QA shows that the types of documents retrieved are not always conducive to correct answers given existing extraction methods. It is not sufficient to retrieve a relevant document if the answer is difficult to extract from its context. Moreover, the retrieval techniques are often very simple, consisting of extracting keywords from questions, expanding them using conventional methods such as synonym expansion and inflectional expansion, and then running the queries through a retrieval engine.

In order to improve overall question answering performance, *additional* documents and *better* documents need to be retrieved. More explicitly, information retrieval needs to: a) generate query types and query content that is designed to be successful (high precision) for individual questions and b) en-

♠ work done at Carnegie Mellon

sure that the documents retrieved by the new queries are different than the documents retrieved using conventional methods. By improving retrieval along these dimensions, we provide QA systems with additional new documents, increasing the diversity and the likelihood of extracting correct answers. In this paper, we present a cluster-based method for expanding queries with new content learned from the process of answering similar questions. The new queries are very different from existing content since they are not based on the question being answered, but on content learned from other questions.

1.1 Related Work

Experiments using the CMU Javelin (Collins-Thompson et al., 2004) and Waterloo's MultiText (Clarke et al., 2002) question answering systems corroborate the expected direct correlation between improved document retrieval performance and QA accuracy across systems. Effectiveness of the retrieval component was measured using *question coverage* – number of questions with at least one relevant document retrieved – and *mean average precision*. Results suggest that retrieval methods adapted for question answering which include question analysis performed better than ad-hoc IR methods which supports previous findings (Monz, 2003).

In question answering, queries are often ambiguous since they are directly derived from the question keywords. Such query ambiguity has been addressed in previous research (Raghavan and Allan, 2002) by extracting part of speech patterns and constructing clarification queries. Patterns are mapped into manually generated clarification questions and presented to the user. The results using the *clarity* (Croft et al., 2001) statistical measure suggest that query ambiguity is often reduced by using clarification queries which produce a focused set of documents.

Another research direction that tailors the IR component to question answering systems focuses on query formulation and query expansion (Woods et al., 2001). Taxonomic conceptual indexing system based on morphological, syntactic, and semantic features can be used to expand queries with inflected forms, hypernyms, and semantically related terms. In subsequent research (Bilotti et al., 2004), stemming is compared to query expansion using inflectional variants. On a particular question answering controlled dataset, results show that expansion using inflectional variants produces higher recall than stemming.

Recently (Riezler et al., 2007) used statistical machine translation for query expansion and took a step towards bridging the lexical gap between questions and answers. In (Terra et al., 2005) query expansion is studied using lexical affinities with different query formulation strategies for passage retrieval. When evaluated on TREC datasets, the affinity replacement method obtained significant improvements in precision, but did not outperform other methods in terms of recall.

2 Cluster-Based Retrieval for QA

In order to explore retrieval under question answering, we employ a statistical system (SQA) that achieves good factoid performance on the TREC QA task: for $\sim 50\%$ of the questions a correct answer is in the top highest confidence answer. Rather than manually defining a complete answering strategy – the type of question, the queries to be run, the answer extraction, and the answer merging methods – for each type of question, SQA learns different strategies for different types of similar questions SQA takes advantage of similarity in training data (questions and answers from past TREC evaluations), and performs question clustering. Two methods are employed constraint-based clustering and EM with similar performance. The features used by SQA clustering are surface-form n-grams as well as part of speech n-grams extracted from questions. However, any clustering method can be employed in conjunction with the methods presented in this paper.

The questions in each cluster are similar in some respect (i.e. surface form and syntax), SQA uses them to learn a complete answering strategy. For each cluster of training questions, SQA learns an answering strategy. New questions may fall in more than one cluster, so multiple answering strategies attempt simultaneously to answer it.

In this paper we do not cover a particular question answering system such as SQA and we do not examine the whole QA process. We instead focus on improving retrieval performance using a set of

similar questions. The methods presented here can generalize when similar training questions are available. Since in our experiments we employ a cluster-based QA system, we use individual clusters of similar questions as local training data for learning better queries.

2.1 Expansion Using Individual Questions

Most existing question answering systems use IR in a simple, straight-forward fashion: query terms are extracted online from the test question and used to construct basic queries. These queries are then expanded from the original keyword set using statistical methods, semantic, and morphological processing. Using these enhanced queries, documents (or passages) are retrieved and the top K are further processed. This approach describes the traditional IR task and does not take advantage of specific constraints, requirements, and rich context available in the QA process. Pseudo-relevance feedback is often used in question answering in order to improve the chances of retrieving relevant documents. In web-based QA, often systems rely on retrieval engines to perform the keyword expansion. Some question answering systems associate additional predefined structure or content based on the question classification. However, there this query enhancement process is static and does not use the training data and the question answering context differently for individual questions.

Typical question answering queries used in document or passage retrieval are constructed using morphological and semantic variations of the content words in the question. However, these expanded queries do not benefit from the underlying structure of the question, nor do they benefit from available training data, which provides similar questions that we already know how to answer.

2.2 Expansion Based on Similar Questions

We introduce cluster-based query expansion (CBQE), a new task-oriented method for query expansion that is complementary to existing strategies and that leads to *different* documents which contain correct answers. Our approach goes beyond single question-based methods and takes advantage of high-level correlations that appear in the retrieval process for similar questions.

The central idea is to cluster available training questions and their known correct answers in order to exploit the commonalities in the retrieval process. From each cluster of similar questions we learn a different, *shared* query content that is used in retrieving relevant documents - documents that contain correct answers. This method leverages the fact that answers to similar questions tend to share contextual features that can be used to enhance keyword-based queries. Experiments with question answering data show that our expanded queries include a different type of content compared to and in addition to existing methods. These queries have training question clusters as a source for expansion rather than an individual test question. We show that CBQE is conducive to the retrieval of relevant documents, *different* than the documents that can be retrieved using existing methods.

We take advantage of the fact that for similar training questions, good IR queries are likely to share structure and content features. Such features can be learned from training data and can then be applied to new similar questions. Note that some of these features cannot be generated through simple query expansion, which does not takes advantage of successful queries for training questions. Features that generate the best performing queries across an entire cluster are then included in a cluster-specific feature set, which we will refer to as the *query content model*.

While pseudo-relevance feedback is performed on-line for each test question, cluster-based relevance feedback is performed across all training questions in each individual cluster. Relevance feedback is possible for training data, since correct answers are already known and therefore document relevance can be automatically and accurately assessed.

Algorithm 1 shows how to learn a query content model for each individual cluster, in particular: how to generate queries enhanced with cluster-specific content, how to select the best performing queries, and how to construct the query content model to be used on-line.

Initially, simple keyword-based queries are formulated using words and phrases extracted directly from the *free* question keywords that do not appear in the cluster definition. The keyword queries are

Algorithm 1 Cluster-based relevance feedback algorithm for retrieval in question answering

1: extract keywords from training questions in a cluster and build keyword-based queries; apply traditional query expansion methods
2: **for all** keyword-based query **do**
3: retrieve an initial set of documents
4: **end for**
5: classify documents into relevant and non-relevant
6: select top k most discriminative features (e.g. n-grams, paraphrases) from retrieved documents (across all training questions).
7: use the top k selected features to enhance keyword-based queries – adding one feature at a time (k new queries)
8: **for all** enhanced queries **do**
9: retrieve a second set of documents
10: **end for**
11: classify documents into relevant and non-relevant based
12: score enhanced queries according to relevant document density
13: include in the *query content model* the top h features whose corresponding enhanced queries performed best across all training questions in the cluster – up to 20 queries in our implementation

then subjected to frequently used forms of query expansion such as inflectional variant expansion and semantic expansion (table **??**). Further processing depends on the available and desired processing tools and may generate variations of the original queries: morphological analysis, part of speech tagging, syntactic parsing. Synonym and hypernym expansion and corpus-based techniques can be employed as part of the query expansion process, which has been extensively studied (Bilotti et al., 2004).

The cluster-based query expansion has the advantage of being orthogonal to traditional query expansion and can be used in addition to pseudo-relevance feedback. CBQE is based on context shared by similar training questions in each cluster, rather than on individual question keywords. Since cluster-based expansion relies on different features compared to traditional expansion, it leads to new relevant documents, different from the ones retrieved using the existing expansion techniques.

3 The Query Content Model

Simple queries are run through a retrieval engine in order to produce a set of potentially relevant documents. While this step may produce relevant documents, we would like to construct more focused queries, likely to retrieve documents with correct answers and appropriate contexts. The goal is to add query content that increases retrieval performance on training questions. Towards this end, we evaluate the discriminative power of features (n-grams and paraphrases), and select the ones positively correlated with relevant documents and negatively correlated with non-relevant documents. The goal of this approach is to retrieve documents containing simple, high precision answer extraction patterns. Features

Cluster: When did **X** start working for **Y**?

Simple Queries	*Query Content Model*
X, **Y**	"**X** joined **Y** in"
X, **Y**, start, working	"**X** started working for **Y**"
X, **Y**, "start working"	"**X** was hired by **Y**"
X, **Y**, working	"**Y** hired **X**"
...	**X**, **Y**, "job interview"
	...

Table 1: Sample cluster-based expansion features

that best discriminate passages containing correct answers from those that do not, are selected as potential candidates for enhancing keyword-based queries. For each question-answer pair, we generate enhanced queries by individually adding selected features (e.g. Table 1) to simple queries. The resulting queries are subsequently run through a retrieval engine and scored using the measure of choice (e.g. average precision). The content features used to construct the top h features and corresponding enhanced queries are included in the *query content model*.

The *query content model* is a collection of features used to enhance the content of queries which are successful across a range of similar questions (Table 1). The collection is *cluster specific* and not *question specific* - i.e. features are derived from training data and enhanced queries are scored using training question answer pairs. Building a query content model does not preclude traditional query expansion. Through the query content model we allow shared context to play a more significant role in query generation.

4 Experiments With Cluster-Based Retrieval

We tested the performance of cluster-based content enhanced queries and compared it to the per-

formance of simple keyword-based queries and to the performance of queries expanded through synonyms and inflectional variants. We also experiment with several feature selection methods for identifying content features conducive to successful queries.

These experiments were performed with a web-based QA system which uses the Google API for document retrieval and a constraint-based approach for question clustering. Using this system we retrieved $\sim 300,000$ and built a document set of $\sim 10GB$. For each new question, we identify training questions that share a minimum surface structure (e.g. a size 3 skip-ngram in common) which we consider to be the prototype of a loose cluster. Each cluster represents a different, implicit notion of question similarity based on the set of training questions it covers. Therefore different clusters lead to different retrieval strategies. These retrieval experiments are restricted to using only clusters of size 4 or higher to ensure sufficient training data for learning queries from individual clusters. All experiments were performed using leave-one-out cross validation.

For evaluating the entire statistical question answering system, we used all questions from TREC8-12. One of the well-known problems in QA consists of questions having several unknown correct answers with multiple answer forms – different ways of expressing the same answer. Since we are limited to a set of answer keys, we avoid the this problem by using all temporal questions from this dataset for evaluating individual stages in the QA process (i.e. retrieval) and for comparing different expansion methods. These questions have the advantage of having a more restrictive set of possible answer surface forms, which lead to a more accurate measure of retrieval performance. At the same time they cover both more difficult questions such as "*When was General Manuel Noriega ousted as the leader of Panama and turned over to U.S. authorities?*" as well as simpler questions such as "*What year did Montana become a state?*". We employed this dataset for an in-depth analysis of retrieval performance.

We generated four sets of queries and we tested their performance. We are interested in observing to what extent different methods produce additional relevant documents. The initial set of queries are constructed by simply using a bag-of-words approach on the question keywords. These queries are run through the retrieval engine, each generating 100 documents. The second set of queries builds on the first set, expanding them using synonyms. Each word and potential phrase is expanded using synonyms extracted from WordNet synsets. For each enhanced query generated, 100 documents are retrieved. To construct the third set of queries, we expand the queries in the first two sets using inflectional variants of all the content words (e.g. verb conjugations and noun pluralization (Bilotti et al., 2004)). For each of these queries we also retrieve 100 documents.

When text corpora are indexed without using stemming, simple queries are expanded to include morphological variations of keywords to improve retrieval and extraction performance. Inflectional variants include different pluralizations for nouns (e.g. *report, reports*) and different conjugations for verbs (e.g. *imagine, imagines, imagined, imagining*). Under local corpus retrieval inflectional expansion bypasses the unrelated term conflation problem that stemmers tend to have, but at the same time, recall might be lowered if not all related words with the same root are considered. For a web-based question answering system, the type of retrieval depends on the search-engine assumptions, permissible query structure, query size limitation, and search engine bandwidth (allowable volume of queries per time). By using inflectional expansion with queries that target web search engines, the redundancy for supporting different word variants is higher, and has the potential to increase answer extraction performance. Finally, in addition to the previous expansion methods, we employ our cluster-based query expansion method. These queries incorporate the top most discriminative ngrams and paraphrases (section 4.1) learned from the training questions covered by the same cluster. Instead of further building an expansion using the original question keywords, we expand using contextual features that co-occur with answers in free text. For all the training questions in a cluster, we gather statistics about the co-occurrence of answers and potentially beneficial features. These statistics are then used to select the best features and apply them to new questions whose answers are unknown. Figure 1 shows that approx-

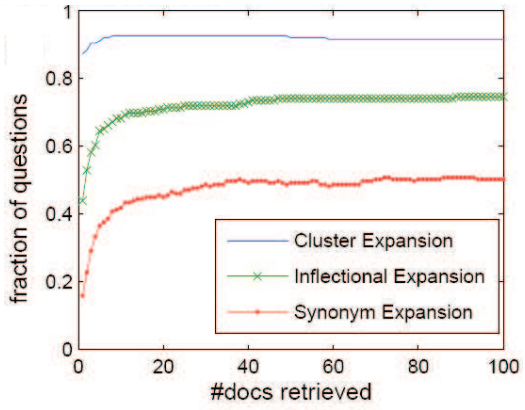


Figure 1: Cumulative effect of expansion methods

imately 90% of the questions **consistently** benefit from cluster-based query expansion when compared to approximately 75% of the questions when employing the other methods combined. Each question can be found in multiple clusters of different resolution. Since different clusters may lead to different selected features, questions benefit from multiple strategies and even though one cluster-specific strategy cannot produce relevant documents, other cluster-specific strategies may be able to.

The cluster-based expansion method can generate a large number of contextual features. When comparing feature selection methods, we only select the top 10 features from each method and use them to enhance existing question-based queries. Furthermore, in order to retrieve, process, extract, and score a manageable number of documents, we limited the retrieval to 10 documents for each query. In Figure 1 we observe that even as the other methods retrieve more documents, $\sim$ 90% of the questions still benefit from the cluster-based method. In other words, the cluster-based method generates queries using a different type of content and in turn, these queries retrieve a different set documents than the other methods. This observation is true even if we continue to retrieve up to 100 documents for simple queries, synonym-expanded queries, and inflectional variants-expanded queries.

This result is very encouraging since it suggests that the answer extraction components of question answering systems are exposed to a different type of relevant documents, previously inaccessible to them. Through these new relevant documents, cluster-based query expansion has the potential to provide answer extraction with richer and more varied sources of correct answers for 90% of the questions.

	new relevant documents	
simple	4.43	100%
synonyms	1.48	33.4%
inflect	2.37	53.43%
cluster	1.05	23.65%
all	9.33	210.45%
all - synonyms	7.88	177.69%
all - inflect	6.99	157.69%
all - cluster	8.28	186.80%

Table 2: Keyword-based ('simple'), synonym, inflectional variant, and cluster-based expansion. Average number of new relevant documents across instances at 20 documents retrieved.

Although expansion methods generate additional relevant documents that simpler methods cannot obtain, an important metric to consider is the density of these new relevant documents. We are interested in the number/percentage of new relevant documents that expansion methods contribute with. Table 2 shows at retrieval level of twenty documents how different query generation methods perform. We consider keyword based methods to be the baseline and add synonym expanded queries ('synonym'), inflectional variants expanded queries ('inflect') which build upon the previous two types of queries, and finally the cluster enhanced queries ('cluster') which contain features learned from training data. We see that inflectional variants have the most impact on the number of new documents added, although synonym expansion and cluster-based expansion also contribute significantly.

4.1 Feature Selection for CBQE

Content features are learned from the training data based on observing their co-occurrences with correct answers. In order to find the most appropriate content features to enhance our cluster-specific queries, we have experimented with several feature selection methods (Yang and Pederson, 1997): information gain, chi-square, and scaled chi-square (phi). Information gain (IG) measures the reduction in entropy for the pre presence/absence of an answer in relevant passages, given an n-gram feature. Chi-square (χ^2) is a non-parametric measure of associa-

tion that quantifies the passage-level association between n-gram features and correct answers.

Given any of the above methods, individual n-gram scores are combined at the cluster level by averaging over individual questions in the cluster. In figure 2 we compare these feature selection methods on our dataset. The selected features are used to enhance queries and retrieve additional documents. We measure the fraction of question instances for which enhanced queries obtain at least one new relevant document. The comparison is made with the document set generated by keyword-based queries, synonym expansion, and inflectional variant expansion. We also include in our comparison the combination of all feature selection methods ('All'). In

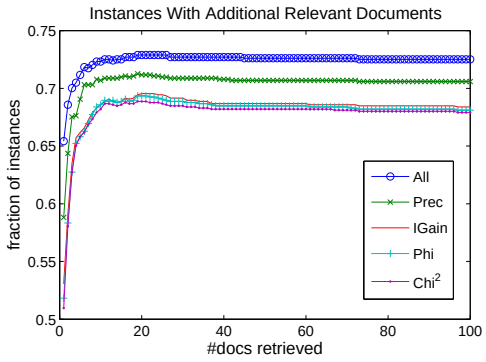


Figure 2: Selection methods for cluster-based expansion

this experiment, average precision on training data proves to be the best predictor of additional relevant documents: $\sim$71% of the test questions benefit from queries based on average precision feature selection. However, the other feature selection methods also obtain a high performance, benefiting $\sim$68% of the test question instances.

Since these feature selection methods have different biases, we expect to observe a boost in performance (73%) from merging their feature sets (Figure 2). In this case there is a trade-off between a 2% boost in performance and an almost double set of features and enhanced queries. This translates into more queries and more documents to be processed. Although it is not the focus of this research, we note that a clever implementation could incrementally add features from the next best selection method only after the existing queries and documents have been processed. This approach lends itself to be a good basis for utility-based models and planning (Hiyakumoto et al., 2005). We in-

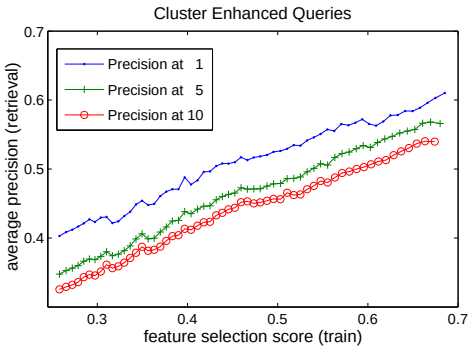


Figure 3: Average precision of cluster enhanced queries

vestigate to what extent the scores of the selected features are meaningful and correlate with actual retrieval performance on test data by measuring the average precision of these queries at different number of documents retrieved. Figure 3 shows precision at one, five, and ten documents retrieved. We observe that feature scores correlate well with actual retrieval performance, a result confirmed by all three retrieval levels, suggesting that useful features learned. The average precision also increases with more documents retrieved, which is a desirable quality in question answering.

4.2 Qualitative Results

The cluster-based relevance feedback process can be used to discover several artifacts useful in question answering. For several of the clusters, we observe that the feature selection process consistently and with high confidence selected features such as "*noun NP1 has one meaning*" where $NP1$ is the first noun phrase in the question. The goal is to add such features to the keyword-based queries to retrieve high precision documents. Note that our example, *NP1* would be different for different test questions.

The indirect reason for selecting such features is in fact the discovery of *authorities*: websites that follow a particular format and which have a particular type of information, relevant to a cluster. In the example above, the websites *answers.com* and *wordnet.princeton.edu* consistently included answers to clusters relevant to a person's biography. Similarly, *wikipedia.org* often provides answers to definitional questions (e.g. "*what is uzo?*"). By includ-

ing non-intuitive phrases, the expansion ensures that the query will retrieve documents from a particular authoritative source – during feature selection, these authorities supplied high precision documents for all training questions in a particular cluster, hence features specific to these sources were identified.

Q: When did Bob Marley die? [A: answers.com]
The noun Bob Marley has one meaning:
Jamaican singer who popularized reggae (1945-81)
Born: *6 February 1945*
Birthplace: *St. Ann's Parish, Jamaica*
Died: *11 May 1981 (cancer)*
Songs: *Get Up, Stand Up, Redemption Song ...*

In this example, profiles for many entities mentioned in a question cluster were found on several *authority* websites. Due to unlikely expansions such as "*noun Bob Marley has one meaning*" the entity "Bob Marley", the answer to the question "*When did Bob Marley die?*" can easily be found. In fact, this observation has the potential to lead to a cluster-based authority discovery method, in which certain sources are given more credibility and are used more frequently than others. For example, by observing that for most questions in a cluster, the *wikipedia* site covers at least one correct answer (ideally that can actually be extracted), then it should be considered (accessed) for test questions before other sources of documents. Through this process, given a set of questions processed using the IBQA approach, a set of authority answer sources can be identified.

5 Conclusions & Future Work

We presented a new, cluster-based query expansion method that learns query content which is successfully used in answering other similar questions. Traditional QA query expansion is based only on the individual keywords in a question. In contrast, the cluster-based expansion learns features from context shared by similar training questions from a cluster.

Since the features of cluster-based expansion are different from the features used in traditional query expansion, they lead to new relevant documents that are different from documents retrieved using existing expansion techniques. Our experiments show that more than 90% of the questions benefit from our cluster-based method when used in addition to traditional expansion methods.

Retrieval in local corpora offers more flexibility in terms of query structure and expressivity. The cluster-based method can be extended to take advantage of structure in addition to content. More specifically, different query structures could benefit different types of questions. However, learning structure might require more training questions for each cluster. Further research can also be done to improve the methods of combining learned content into more robust and generalizable queries. Finally we are interested modifying our cluster-based expansion for the purpose of automatically identifying authority sources for different types of questions.

References

M. W. Bilotti, B. Katz, and J. Lin. 2004. What works better for question answering: Stemming or morphological query expansion? In *IR4QA, SIGIR Workshop*.

C. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, and P. Tilker. 2002. Statistical selection of exact answers.

K. Collins-Thompson, E. Terra, J. Callan, and C. Clarke. 2004. The effect of document retrieval quality on factoid question-answering performance.

W.B. Croft, S. Cronen-Townsend, and V. Lavrenko. 2001. Relevance feedback and personalization: A language modeling perspective. In *DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*.

L. Hiyakumoto, L.V. Lita, and E. Nyberg. 2005. Multi-strategy information extraction for question answering.

C. Monz. 2003. From document retrieval to question answering. In *Ph. D. Dissertation, Universiteit Van Amsterdam*.

H. Raghavan and J. Allan. 2002. Using part-of-speech patterns to reduce query ambiguity.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *ACL*.

E. Terra, C.L., and A. Clarke. 2005. Comparing query formulation and lexical affinity replacements in passage retrieval. In *ELECTRA, SIGIR Workshop*.

W.A. Woods, S.J. Green, P. Martin, and A. Houston. 2001. Aggressive morphology and lexical relations for query expansion.

Y. Yang and J. Pederson. 1997. Feature selection in statistical learning of text categorizatio n.

A Semantic Feature for Relation Recognition Using A Web-based Corpus

Chen-Ming Hung
Institute of Information Science
Academia Sinica, Taipei, Taiwan
rglly@iis.sinica.edu.tw

Abstract

Selecting appropriate features to represent an entity pair plays a key role in the task of relation recognition. However, existing syntactic features or lexical features cannot capture the interaction between two entities because of the dearth of annotated relational corpus specialized for relation recognition. In this paper, we propose a semantic feature, called the $latent\ topic\ feature$, which is topic-based and represents an entity pair at the semantic level instead of the word level. Moreover, to address the problem of insufficiently annotated corpora, we propose an algorithm for compiling a training corpus from the Web. Experiment results demonstrate that $latent\ topic\ features$ are as effective as syntactic or lexical features. Moreover, the Web-based corpus can resolve the problems caused by insufficiently annotated relational corpora.

1 INTRODUCTION

Relation recognition is a challenging task because finding appropriate features to represent the relationship between two entities is difficult and limited by the scarcity of annotated corpora. Prior works on relation recognition have focused on syntactic features, e.g., parsing trees (Culotta and Sorensen, 2004; Zelenko et al., 2003), and on lexical features, e.g., Part-Of-Speech (POS) features. These approaches show that syntactic features and lexical features outperform bag-of-words (BOW) on existing annotated corpora such as the RDC corpus of the ACE project. The superior performance achieved by syntactic and lexical features is due to their ability to capture the grammatical relations between two entities and the characteristics of the entities. For example, (Culotta and Sorensen, 2004) add hypernyms of entities to features derived from WordNet. However, neither syntactic nor lexical features can capture the interaction between two entities at the semantic level.

Another issue in the task of relation recognition is insufficiently annotated corpora. For example, given a pair $\{$*the U.N. body, Kosovo*$\}$, we can only find three sentences containing both entities in the RDC corpus, which is commonly used corpus in the relation recognition task. The problem of an insufficiently annotated corpus biases feature vectors and distorts the prediction of entity pairs. However, (Huang et al., 2004; Hung and Chien, 2007) have shown that the Web can be used as an alternative source of documents related to a given query. That is possibly because of the increasing size of the Web and the efficiency in commercial search engines, e.g., Google and Yahoo!.

To resolve the above problems, we propose a semantic feature called the *latent topic feature*, which is extracted by exploiting the *Latent Dirichlet Allocation* (LDA) algorithm. Unlike syntactic features or lexical features, *latent topic features* represent entity pairs as random mixtures of latent topics, where each topic is characterized by a distribution of words. We prove experimentally that *latent topic features* are as effective as syntactic features or lexical features in capturing the interaction between two entities. The experiment results are predictable. In

the above {*the U.N. body, Kosovo*} example, it may be difficult to determine the relationship between *U.N. body* and *Kosovo* straightforwardly. However, making the right guess about the relationship is easier if *the U.N. body* is grouped with *army* and *government*. Therefore, the $right\ guess$ in this example is $management$.

To overcome the problems caused by an insufficiently annotated corpus, we exploit the Web as a source of training data for the relation recognition task. Given an entity pair, documents describing the entity pair are extracted from the Web via commercial search engines using both entities as the query. In other words, snippets returned from the Web are treated as documents related to the query. Our assumption, which has been proved in previously published works, is that returned snippets can capture the interaction between two entities. After the *latent topic features* extracted from returned snippets using the Web as the corpus, an SVM classifier is trained as the relation recognition classifier for use in the later experiments.

The remainder of the paper is organized as follows. in Section 2, we discuss works related to feature selection in the relation recognition task as well as using the Web as a corpus. The concept of *latent topic features* is presented in Section 3. We also explain how we represent a document in the vector space of a *latent topic feature*. Section 4contains an evaluation of the *latent topic feature*. We then present our conclusions in Section 5.

2 RELATED WORK

In the field of information extraction (IE), the goal of relation recognition is to find the relationship between two entities. Without considering entity detection, relation recognition depends heavily on the representation of entity pairs. (Zelenko et al., 2003) showed how to extract relations by computing the kernel functions between the kernels of shallow parse trees. The kernels are defined over a shallow parse representation of the text and used in conjunction with a Support Vector Machine (SVM) learning algorithm to extract person-affiliation and organization-location relations. (Culotta and Sorensen, 2004) extended this work to estimate kernel functions between augmented dependency trees, while (Kambhatla, 2004) combined lexical features, syntactic features, and semantic features in a maximum entropy model. However, the semantic features discussed in (Kambhatla, 2004) still focus on the word level instead of the conceptual level.

LDA is an aspect model that represents documents as a set of $topics$ instead of a bag-of-words. *Latent semantic indexing (LSI)* (Deerwester et al., 1990) and *probabilistic latent semantic indexing (PLSI)* (Hofmann, 1999) are also aspect models and have been widely used in the field of information retrieval. LSI simply assumes that each document is generated from single latent topic, while PLSI attempts to relax the assumption by using a mixture of latent topics for each document. However, PLSI is highly dependent on training documents; in other words, it cannot handle the probability of latent topics in a previously unseen document. In addition, the number of parameters that must be estimated in PLSI grows linearly with the number of training documents. (Blei et al., 2003; Blei and Jordan, 2003) proposed LDA to resolve the abovementioned limitations. It can easily generate an unseen document under controllable parameters.

A number of works, e.g., (Huang et al., 2004), have investigated using the Web to acquire a training corpus or acquire additional information not provided by existing annotated corpora. (Huang et al., 2004) exploited the Web as a training corpus to train a classifier with user-defined categories. However, it is widely recognized that when using documents on the Web users must spend a great deal of time filtering out unrelated contents. (Hung and Chien, 2007) designed a bootstrapping method that adapts an existing corpus with an automatic verification algorithm in order to control the quality of returned snippets in each iteration. (Matsuo et al., 2006) used the Web to construct a social network system, called *POLYPHONET*, which visualizes the relationship between two personal names.

3 LATENT TOPIC FEATURE

In this section, we introduce the concept of using the Web to augment an insufficiently annotated corpus for relation recognition. Then we apply the LDA algorithm to the corpus to extract the $latent\ topic$

$features$ to represent entity pairs in the corpus for relation recognition.

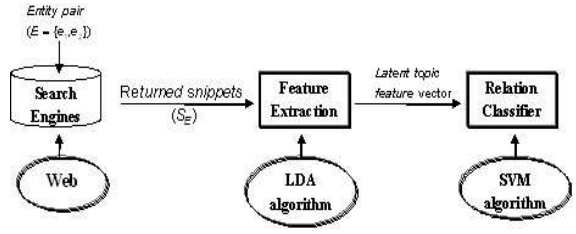


Figure 1: The framework of the proposed approach.

3.1 Compiling a Web-based Relational Corpus

For an entity pair, $E = \{e_1, e_2\}$, where e_1 and e_2 are named entities, it is difficult to find sufficient sentences to describe their relationship from existing annotated corpora. In other words, given an entity pair without a relation label, users cannot recognize the pair. Even with a widely used thesaurus, like WordNet, we can only obtain hypernyms or synonyms of given entities. It is not possible to obtain knowledge about the interaction between two entities.

To capture the interaction between two entities, we send both entities, $e_1\ AND\ e_2$, to commercial search engines and collect returned snippets as training documents for an entity pair, E. Snippets of returned search results are defined as the surrounding contexts of queries highlighted by commercial search engines. In other words, the full texts of search results are not considered in the collected corpus when filtering noisy information in full documents. Let R be the relation label of entity pairs $\{E_1, \ldots, E_M\}$; then, the training corpus for R is the collection of all returned snippets for $\{E_1, \ldots, E_M\}$. Through effective commercial search engines such as Google and Yahoo!, sentences describing the interaction between two entities can easily be retrieved. Returning to the example $\{$*the U.N. body, Kosovo*$\}$, almost two million sentences with co-occurrences of the two entities are retrieved by Google. Another advantage of using the Web to retrieve relevant documents is the $auto-correction$ ability of commercial search engines. The feature can correct a misspelled query or replace an uncommon word with a synonym or a common word which is correct, so that more related information about entity pairs can be retrieved from the Web as returned snippets. For example, Google can automatically link $the\ U.N.\ body$ to $United\ Nations$, which is used more frequently in searching. Clearly, the number of returned snippets must be considered. Actually, based on experiment results in Section 4, we set the number as five, which achieves the best performance.

3.2 Modified LDA for the Relational Corpus

LDA is an aspect model with three levels, namely, the corpus level, the document level, and the word level. Given a document, variables of the corpus it belongs to are sampled first, after which the variable of the document is sampled once. Finally, variables for words in the document are sampled.

For a document d in a corpus D, the modeling process is as follows:

1. Sample $\theta \sim Dir(\theta|\alpha)$.
2. For each word w_n in d, $n \in \{1, \ldots, N\}$:
   (a) sample $z_n \sim Mult(\theta)$,
   (b) sample a word $w_n \sim p(w_n|z_n, \beta)$ from a multivariate Gaussian distribution conditioned on the topic z_n.

Note that α is a vector of corpus-level variables whose dimensionality is equal to the number of latent topics; θ is a variable of the document and is assumed to follow Dirichlet distribution for the given corpus; and β is a word-level variable. In addition, $Z = \{z_1, z_2, \ldots, z_N\}$ are latent factors that generate the document, and z_n is the latent topic that w_n is generated from. Finally, N is the length of the document d.

An entity pair E in the relational corpus is similar to a document d in the text corpus. In other words, the corpus D^R is comprised of returned snippets for all entity pairs E^R with the same relation label R. Therefore, given the parameters α and β, we obtain the distribution of entity pair E as follows:

$$p(E|\alpha, \beta) = \int \sum_{z_n} p(\theta, z_n, S_E|\alpha, \beta) d\theta,$$

where

$$p(\theta, z_n, S_E|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N_E} p(z_n|\theta) p(w_n|z_n, \beta),$$

N_E is the number of words in the returned snippets for E; and w_n is the n^{th} word in S_E, the returned snippets of E. Table 1 summarizes notations used in the paper.

Table 1: Notations used in this paper.

SYMBOL	DESCRIPTION
R	relation label
D^R	corpus for R
E_j^R	j^{th} entity pair in the relation label R
S_E	returned snippets for an entity pair E
$\|E^R\|$	number of entity pairs in the relation R
N_E	number of words in S_E
w_n	n_{th} word in S_E
z_n	latent topic that w_n is generated from

In Section 3.1, we discussed the advantages of using the Web as a corpus to model entity pairs. In the modeling process, we estimate the probability of w_n conditioned on z_n, $p(w_n|z_n, \beta)$, to maximize the probability of the entire corpus of R. The probability that we try to maximize is

$$p(D^R|\alpha, \beta) = \prod_{j=1}^{|E^R|} p(E_j^R|\alpha, \beta).$$

3.3 Latent Topic Feature

In different corpora, z obtains a different distribution to maximize the likelihood of the given corpus. In this section, we describe how to exploit z as features to represent a snippet for an entity pair E, i.e., S_E. In Section 3.2, we noted that the parameters to be estimated in the aspect model are all probabilities of words in each latent topic z. Thus, we let the expected number of words generated from latent topics be features of each entity pair. In other words, an entity pair E is represented as a feature vector whose length is equal to the number of latent topics and whose i^{th} attribute is equal to

$$\alpha_i + \sum_{n=1}^{N_E} |w_n| \times p(w_n|z_n, \beta),$$

where α_i is the i^{th} prior Dirichlet parameter.

In addition, because there is no solution good enough to determine the dimensionality of the feature vector or the number of latent topics, we set the number of topics at thirty because it probably minimize the computation cost without significantly affecting the performance.

4 EXPERIMENTS

In this section, we evaluate the performance of the $latent\ topic\ feature$ in representing entity pairs extracted from the Relation Detection and Characterization (RDC) corpus of the Automatic Content Extraction 2003 model (ACE 2003)[1].

4.1 The RDC Corpus

In the RDC corpus, five relation types, AT, $NEAR$, $PART$, $ROLE$, and SOC, are defined; each relation type has extended sub-relations. Table 2 summarizes the relations in the RDC corpus for ACE 2003. Based on Table 2, we find that the distribution of the number relations is very unbalanced, ranging from 2 to 773. In the following experiments, we only consider the $Role$ relation because it has the largest numbers of sub-relations and it is easier to verify the recognition results manually. Note that a relation is dropped if it has less than ten sub-relations in order to avoid the bias of learned classifiers. Therefore, the sub-relation $founder$ in $Role$ is dropped in the following experiments because it occurs less than ten times. $Other$ is also dropped because its definition is unclear.

Table 2: Distribution over relation types in the RDC corpus (ACE 2003).

Relations	Sub-Relations(Size)	
AT	Based-in(78)	Located(773)
	Residence(186)	
NEAR	Relative-location(73)	
PART	Part-of(242)	Subsidiary(172)
	Other(2)	
ROLE	Affiliate-partner(34)	
	Citizen-of(93)	Client(33)
	Founder(6)	General-Staff(460)
	Management(294)	Member(398)
	Owner(41)	Other(98)
SOC	Associate(25)	Grandparent(3)
	Parent(23)	Sibling(5)
	Spouse(22)	Other-relative(24)
	Other-Personal(10)	
	Other-Professional(88)	

[1] http://projects.ldc.upenn.edu/ace/

4.2 Setting and Measurement

We used the package of (Chang and Lin, 2001) to design the following experiments. In addition, ν-SVM with a radial kernel function was used to learn the relation classifier. To determine the parameters in ν-SVM, i.e., γ and ν, we observed the performance of the ν-SVM classifier by randomly selecting 80% of the sentences in the RDC corpus as training data and the remaining 20% as test data. In other words, we applied five-fold cross validation to build a temporary model for parameter estimation. Furthermore, it is well known that parameters in the SVM model must be optimized manually; therefore, we estimate ν first and then estimate γ. γ is fixed while ν is being estimated and vice versa. After estimation, the best result is achieved at the point that γ is equal to 2.5×10^{-4} and ν is equal to 0.05. We summarize the results in Figure 2. The top graph in Figure 2 is the accuracy curve, where fixed $\gamma = 2.5 \times 10^{-4}$ and flexible ν; the bottom graph is the accuracy curve with fixed $\nu = 0.05$ and flexible γ.

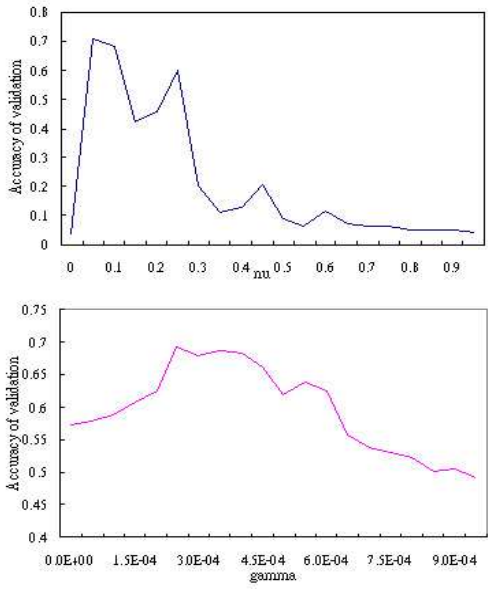


Figure 2: Accuracy of five-fold cross validation using bigram features. Top: ν with $\gamma = 2.5 \times 10^{-4}$. Bottom: γ with $\nu = 0.05$.

For each sub-relation in $Role$, binary classification is used in the experiments and the F-measure of each sub-relation is used as the metric for assessing the performance of $latent\ topic\ features$.

$$
\begin{aligned}
F - value &= \frac{2 \times Precision \times Recall}{Precision + Recall} \\
Recall &= \frac{\sharp \text{ of correct positive predictions}}{\sharp \text{ of positive examples}} \\
Precision &= \frac{\sharp \text{ of correct positive predictions}}{\sharp \text{ of positive predictions}}
\end{aligned}
$$

4.3 Web-based Corpus vs. Annotated Corpus

We now evaluate the performance of relational classifier on a Web-based corpus and on an annotated corpus. To assess the performance of on the annotated corpus, sentences in the RDC corpus containing a co-occurrence of both given entities were extracted as training data to learn a benchmark relation classifier. On the other hand, the Web-based corpus is compiled from snippets retrieved by using both entities as a query. The $latent\ topic\ feature$ is applied on both the Web-based corpus and the RDC corpus using the procedure described in Section 4.2. In addition, to analyze the effect of the number of returned snippets, we increased the number of snippets from 3 to 45 in increments of three and then summarized the relationship between the number of returned snippets and the achieved accuracy curve shown in Figure 3. In the figure, the training data is comprised of snippets of information returned by querying 80% of the entity pairs selected at random in the RDC corpus. The test data comprises snippets returned by querying the remaining 20% of entity pairs in the corpus.

From Figure 3, we observe that using five returned snippets for each entity pair achieves the best accuracy (0.85), which is substantially higher than the accuracy achieved by using annotated corpus (0.69). Note that using more returned snippets does not guarantee higher accuracy. For example, when 39 returned snippets are used for each entity pair, the accuracy (0.56) is almost the same as that (0.55) achieved by using only 3 returned snippets. Moreover, it is significantly less than the accuracy (0.69) achieved by using the RDC corpus. This is reasonable because the greater the number of returned snippets, the larger the amount of noisy information introduced to the classifier, which degrades its performance.

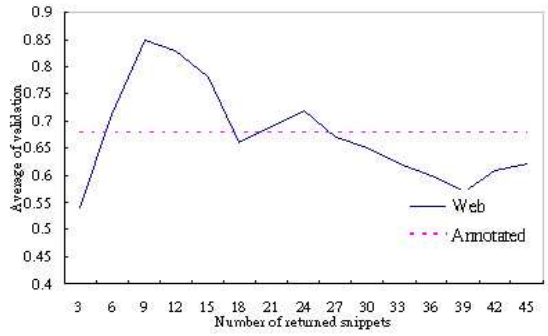


Figure 3: Accuracy of five-fold cross validation using the Web-based corpus and the annotated corpus.

4.4 Latent Topic Feature vs. Other Features

In this section, we compare the performance of $latent\ topic\ features$ with that of syntactic features and lexical features, i.e., $bag - of - words$ or $parts - of - speech$. Because of the superior performance achieved by using the Web-based corpus described in Section 4.3, we extracted features from the training corpus compiled from that corpus rather than the annotated corpus.

Based on the results reported in Section 4.3, five snippets were returned by the Web-based corpus for each entity pair. For each sub-relation, a one-class SVM was trained to perform binary classification.

Each sub-relation of $Role$ in Table 3 is applied with binary classification using a one-class SVM. Table 5 summarizes the results of a comparison between the $latent\ topic\ feature$ and the features used by (Culotta and Sorensen, 2004). The latter depends on dependency tree kernels, which represent the grammatical dependencies in a sentence and are considered as syntactical features. In Table 5, BOW denotes bag-of-word, $sparse$ represents a sparse kernel, and $contiguous$ represents a contiguous kernel.

Surprisingly, for every sub sub-relation in Table 3, the $latent\ topic\ feature$ consistently achieves a significantly higher average recall rate, but a lower average precision rate. This may be due to the $latent\ topic\ feature's$ ability to capture information at the semantic level precisely, but it cannot distinguish the information at the word level easily. In other words, the $latent\ topic\ feature$ can capture the common semantic information, probably the $Role$, of all sub-relations, but it cannot tell the difference between $citizen - of$ and $founder$. Table 4 shows the results of applying binary classification to five relations in the RDC corpus. Although the precision rate for each relation is still low, the recall rate has been increased significantly. This demostrates the ability of the $latent\ topic\ feature$ to capture semantic information.

Table 3: Binary classification results for each sub-relation of $Role$.

	Latent Topic Feature		
	F	**Prec.**	**Rec.**
Aff.-Part.	0.30	0.18	1.00
Client	0.40	0.28	0.71
Citizen-Of	0.62	0.47	0.91
Gen.-Staff	0.78	0.64	0.99
Manage.	0.56	0.39	1.00
Member	0.62	0.46	0.93
Owner	0.45	0.29	0.98

Table 4: Binary classification results for each relation in the RDC corpus.

	Latent Topic Feature		
	F	**Prec.**	**Rec.**
At	0.61	0.48	0.84
NEAR	0.36	0.23	0.88
PART	0.58	0.46	0.80
ROLE	0.71	0.64	0.79
SOC	0.59	0.45	0.87

In Table 5, although the recall rate using the $latent\ topic\ feature$ is much higher than that achieved by the other features, unfortunately, the $F - score$ of the $latent\ topic\ feature$ cannot be redeemed because of the much lower precision rate. Moreover, the $latent\ topic\ feature$ is comparable to the sparse kernel method in a different way because it has a low precision rate but a high recall rate. Finally, the $latent\ topic\ feature$ achieves a higher average F-score than the bag-of-words feature, which proves the assumption that the $latent\ topic\ feature$ can better capture the interaction between two entities than features at the word level.

5 CONCLUSION

We have proposed a concept called the $latent\ topic\ feature$ for the task of relation recognition and evaluated it on the RDC of the ACE project. The feature captures the interaction between two entities

Table 5: Comparison between $Latent\ topic\ feature$ and other features.

	Average		
	F	**Prec.**	**Rec.**
$Latent\ Topic$	0.58	0.45	0.84
Sparse	0.59	0.83	0.46
Contiguous	0.62	0.85	0.49
BOW	0.52	0.73	0.40
Sparse+BOW	0.62	0.80	0.50
Cont.+BOW	0.63	0.81	0.52

at the semantic level rather than at the word level. Therefore, combining the $latent\ topic\ feature$ with syntactic features and lexical features should achieve a better performance than using the features separately. In our future work, we will devise an appropriate way of combining $latent\ topic\ features$ with syntactical and lexical features.

Because of the lack of a sufficiently annotated corpus for relation corpus for relation recognition, we have also proposed using a Web-based corpus to train classifiers for the purpose. Our experiment results demonstrates that Web documents can accurately capture information about the interaction between two named entities in the absence of an annotated corpus. By using a Web-based corpus, the time cost to manually annotating a corpus for relation recognition is expected to be significantly reduced if the quality of returned snippetscan be controlled.

References

David M. Blei and Michael I. Jordan. 2003. Modeling annotated data. In *Proceedings of the 26th SIGIR*, pages 127–134. ACM Press.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*.

A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm.

Chien-Chung Huang, Shui-Lung Chuang, and Lee-Feng Chien. 2004. Liveclassifier: creating hierarchical text classifiers through web corpora. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 184–192, New York, NY, USA.

Chen-Ming Hung and Leeq-Feng Chien. 2007. Web-based text classification in the absence of manually labeled training documents. *J. Am. Soc. Inf. Sci. Technol.*, 58(1):88–96.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 178–181, Barcelona, Spain, July. Association for Computational Linguistics.

Yutaka Matsuo, Junichiro Mori, Masahiro Hamasaki, Keisuke Ishida, Takuichi Nishimura, Hideaki Takeda, Koiti Hasida, and Mitsuru Ishizuka. 2006. Polyphonet: an advanced social network extraction system from the web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 397–406, New York, NY, USA.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Multilingual Text Entry using Automatic Language Detection

Yo Ehara and **Kumiko Tanaka-Ishii**
Graduate School of Information Science and Technology, University of Tokyo
13F Akihabara Daibiru, 1-18-13 SotoKanda Chiyoda-ku, Tokyo, Japan
ehara@r.dl.itc.u-tokyo.ac.jp kumiko@i.u-tokyo.ac.jp

Abstract

Computer users increasingly need to produce text written in multiple languages. However, typical computer interfaces require the user to change the text entry software each time a different language is used. This is cumbersome, especially when language changes are frequent.

To solve this problem, we propose TypeAny, a novel front-end interface that detects the language of the user's key entry and automatically dispatches the input to the appropriate text entry system. Unlike previously reported methods, TypeAny can handle more than two languages, and can easily support any new language even if the available corpus is small.

When evaluating this method, we obtained language detection accuracy of 96.7% when an appropriate language had to be chosen from among three languages. The number of control actions needed to switch languages was decreased over 93% when using TypeAny rather than a conventional method.

1 Introduction

Globalization has increased the need to produce multilingual text — i.e., text written in more than one language — for many users. When producing a text in a language other than English, a user has to use text entry software corresponding to the other language which will transform the user's key stroke sequences into text of the desired language. Such software is usually called an *input method engine* (IME) and is available for each widely used language. When producing a multilingual text on a typical computer interface, though, the user has to switch IMEs every time the language changes in a multilingual text. The control actions to choose an appropriate IME are cumbersome, especially when the language changes frequently within the text.

To solve this problem, we propose a front-end interface called TypeAny. This interface detects the language that the user is using to enter text and dynamically switches IMEs. Our system is situated between the user key entry and various IMEs. TypeAny largely frees the user from the need to execute control actions when switching languages.

The production of multilingual text involves three kinds of key entry action:

- actions to enter text
- actions to control an IME[1]
- actions to switch IMEs

Regarding the first and second types, substantial work has been done in the UI and NLP domain, as summarized in (MacKenzie and Tanaka-Ishii, 2007). There has especially been much work regarding Chinese and Japanese because in these languages the number of actions of the second type is closely related to the accuracy of conversion from Romanized transcription to characters in each of these languages, and this directly reflects the capability of the language model used.

[1]When using predictive methods such as completion, or kana-kanji conversion in Japanese, the user has to indicate to the IME when it should predict and which proposed candidate to choose.

In contrast, this paper addresses the question of how to decrease the need for the third type of action. From the text entry viewpoint, this question has received much less attention than the need to reduce the number of actions of the second type. As far as we know, this issue has only been directly addressed by Chen et al. (2000), who proposed integrating English entry into a Chinese input system rather than implementing multilingual input.

Reports on ways to detect a change in the language used are more abundant. (Murthy and Kumar, 2006) studied the language identification problem based on small samples in several Indian languages when machine learning techniques are used. Although they report a high accuracy for the method they developed, their system handles switches between two Indian languages only. In contrast, TypeAny can handle any number of languages mixed within a text.

(Alex, 2005) addresses a related task, called *foreign inclusion detection* (FID). The task is to find foreign (i.e., English) inclusions, such as foreign noun compounds, within monolingual (i.e., German) texts. Alex reported that the use of FID to build a polyglot TTS synthesizer was also considered (Pfister and Romsdorfer, 2003), (Marcadet et al., 2005). Recently, Alex used FID to improve parsing accuracy (Alex et al., 2007). While FID relies on large corpora and lexicons, our model requires only small corpora since it incorporates the transition probabilities of language switching. Also, while FID is specific to alphabetic languages, we made our method language-independent by taking into consideration the inclusion problem at the key entry level.

In the following, we introduce the design of TypeAny, explain its underlying model, and report on our evaluation of its effectiveness.

2 Design of TypeAny

Figure 1 shows an example of text written in English, Japanese and Russian. The strings shown between the lines indicate the Roman transcription of Japanese and Russian words.

With a conventional computer interface, entering the text shown in Figure 1 would require at least six control actions since there are six switches between languages: from English to Japanese and back, and twice from English to Russian and back. Note that such IME switches are also required even when the text consists only of European languages. Each European language has its own font and diacritic system, which are realized by using IMEs.

Some Japanese restaurants offer いくら, or salmon roe.
ikura
Whilst even Japanese think it is a Japanese word,
surprisingly it is a loan word from Russian икра.
ikra
Caviar is also called black икра in Russian.
ikra

Figure 1: Example of Multilingual Text in English, Japanese and Russian

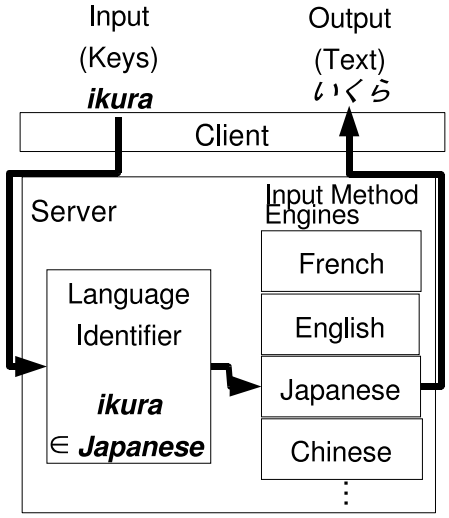


Figure 2: System Structure

TypeAny solves the problem of changing IME. It is situated between the user's key entry and various IMEs as shown in the system architecture diagram of Figure 2. The user's key entry sequence is input to our client software. The client sends the sequence to the server which has the language identifier module. This module detects the language of the key sequence and then sends the key sequence to the appropriate IME[2]. The selected IME then converts the key entries into text of the detected language.

In our study, IMEs for European languages are built using simple transliteration: e.g., "[" typed in an English keyboard is transliterated into "ü" of German. In contrast, the IMEs for Japanese and Chinese

[2]Precisely speaking, TypeAny detects keyboard layouts (i.e., Qwerty, Dvorak, Azerty, etc.) as well as the languages used.

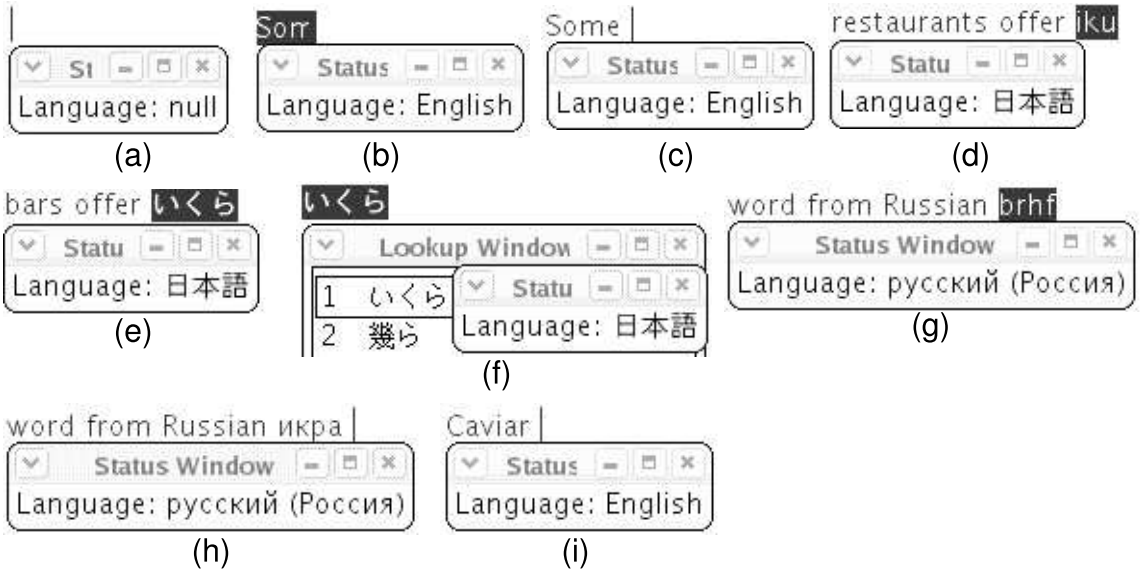


Figure 3: Entry Flow

require a more complicated system because in these languages there are several candidate transcriptions for a key sequence. Fortunately, several existing software resources can be used for this. We use Anthy[3] as the IME for Japanese. As for the IME for Chinese, we used a simple word-based pinyin-hanzi conversion system.

TypeAny restarts the language detection every time a certain delimiter appears in the user's key sequence. By using delimiters, the system can avoid resorting to a combinatorial search to find the border between languages. Such delimiters naturally occur in natural language texts. For example, in the case of European languages, blank spaces are used to delimit words and it is unlikely that two languages will be mixed within one word. In languages such as Chinese and Japanese, blank spaces are typically used to indicate that the entry software should perform conversion, thus guaranteeing that the sequence between two delimiters will consist of only one language[4]. Therefore, assuming that a text fragment between two delimiters is written in just one language is natural for users. A text fragment between two delimiters is called a *token* in TypeAny.

An example of the TypeAny procedure to enter the text from Figure 1[5] is shown in Figure 3. In each step in Figure 3, the text is entered in the first line, where the token that the user is entering is highlighted. The language estimated from the token is shown in the locale window shown below (called the Status Window). Each step proceeds as follows.

(a) The initial state.

(b) The user first wants to type a token "Some" in English. When "Som" is typed, the system identifies that the entry is in English. The user confirms this language by looking at the locale window.

(c) The user finishes entering the token "Some" and when the user enters a blank space, the token "Some" is confirmed as English text. TypeAny restarts detection for the language of the next token. The tokens up to and including "offer" are entered similarly to "Some".

(d) The user types in a token "ikura" in Japanese. The moment "iku" is typed, "iku" is identified as Japanese, as is confirmed by the user through the locale window.

(e) When the user finishes entering the token "ikura" and types in a blank space, the sequence is sent to a Japanese IME to be converted into "ikura", so that a Japanese text fragment is obtained.

(f) Through conventional kana-kanji conversion,

[3] http://anthy.sourceforge.jp/

[4] Note that a token can consist of a sequence longer than a word, since many types of conversion software allow the conversion of multiple words at one time.

[5] This case assumes use of the Qwerty keyboard.

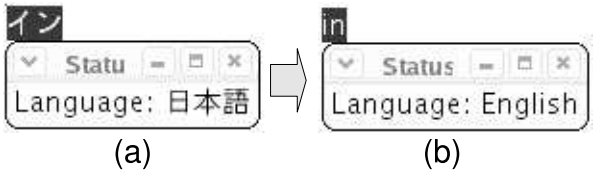


Figure 4: When Detection Fails

the user can select the appropriate conversion of "ikura" among from candidates shown in the Lookup Window and the token is confirmed. TypeAny begins detecting the language of the next token. The tokens between "or" and "Russian" are successfully identified as English in a way similar to procedures (b) and (c).

(g) The key entry "brhf" is the key sequence for the Russian token whose English transliteration is "ikra".

(h) Since "brhf" is identified as Russian, "brhf" is converted into Russian characters.

(i) The following word "Caviar" is detected as English, as in (b) and (c).

As seen in this example, the user does not need to take any action to switch IMEs to enter tokens of different languages.

Two types of detection failure occur in TypeAny:

Failure A: the language should switch, but the new language is incorrectly selected.

Failure B: the language should not switch, but TypeAny misjudges that it should.

While conventional methods require a control action every time the language switches, TypeAny requires a control action *only* to correct such a failure. Therefore, **Failure A** never increases the number of control actions compared to that of conventional methods. On the other hand, **Failure B** errors are a concern as such failures might increase the number of control actions to beyond the number required by a conventional method. Thus, the effectiveness of introducing TypeAny depends on a trade-off between fewer control actions at language switching points and potentially more control actions due to **Failure B** errors. Our evaluation in §4.2 shows that the increase in the number of actions due to **Failure B** errors is insignificant.

In the event of failures, the user can see that there is a problem by watching the locale window and then easily correct the language by pressing the TAB key. For example, while "in" was correctly judged for our example, suppose it is incorrectly detected as Japanese as shown in Figure 4(a). In this case, the user can manually correct the locale by pressing the TAB key once. The locale is then changed from Figure 4(a) (where "in" is identified as Japanese), to (b) where "in" is identified as English.

Note that the language of some tokens will be ambiguous. For example, the word "sushi" can be both English and Japanese because "sushi" has almost become an English word: many loan words share this ambiguity. Another case is when diacritic marks are considered: for example, the word "fur" is usually an English word, but some German users may wish to use this word as "für" without diacritic marks. Such a habit is widely seen among users of European languages. Some of this sort of ambiguity is disambiguated by considering the context and by online learning, which is incorporated in the detection model as explained next.

3 Language Detection

3.1 Language Detection Model

We modeled the language detection as a hidden Markov model (HMM) process whose states correspond to languages and whose outputs correspond to tokens from a language.

Here, the goal is to estimate the languages $\hat{l}_1^m$ by maximizing $\mathrm{P}(l_1^m, t_1^m)$, where $l \in L$ denotes a language in L, a set of languages, and t denotes a token[6]. By applying a hidden Markov model, the maximization of $\mathrm{P}(l_1^m, t_1^m)$ is done as shown in Equation (1).

$$\begin{aligned}\hat{l}_1^m &= \operatorname*{argmax}_{l_1^m \in L} \mathrm{P}(l_1^m, t_1^m) \\ &= \operatorname*{argmax}_{l_1^m \in L} \mathrm{P}(t_1^m | l_1^m) \mathrm{P}(l_1^m) \\ &\approx \operatorname*{argmax}_{l_1^m \in L} \left(\prod_{i=1}^{m} P(t_i|l_i) \right) \left(\prod_{i=1}^{m} P(l_i | l_{i-k}^{i-1}) \right) \quad (1)\end{aligned}$$

In the last transformation of Equation (1), it is assumed that $\mathrm{P}(t_1^m|l_1^m) \approx \prod_{i=1}^m P(t_i|l_i)$ and $\mathrm{P}(l_i|l_1^{i-1}) \approx P(l_i|l_{i-k}^{i-1})$ for the first and the second terms, respectively. In Equation (1), the first term

[6]Let $t_u^v = (t_u, t_{u+1}, \ldots, t_v)$ be an ordered list consisting of $v - u + 1$ elements for $v \geq u$.

corresponds to the *output probabilities* and the second term corresponds to the *transition probabilities*.

In a usual HMM process, a system finds the language sequence (i.e., state sequence) l_1^m that maximizes Equation (1) by typically using a Viterbi algorithm. In our case, too, the system can estimate the language sequence for a sequence of tokens. However, as discussed earlier, since it is unlikely that a user enters a token consisting of multiple languages, our system is designed only to estimate the language of the latest token l_m, supposing that the languages of the previous l_1^{m-1} are correct.

In the following two sections, the estimation of each term is explained.

3.2 Output Probabilities

The output probabilities $P(t_i|l_i)$ indicate the probabilities of tokens in a monolingual corpus, and their modeling has been substantially investigated in NLP.

Note that the estimation of $P(t_i|l_i)$ requires monolingual corpora. If the corpora are large, $P(t_i|l_i)$ is estimated from the token frequencies. However, because large corpora are not always available, especially for minor languages, $P(t_i|l_i)$ is estimated using key entry sequence probabilities based on n-grams (with maximum n being n_{max}) as follows:

$$P(t_i|l_i) = P(c_1^{|t_i|}|l_i) = \prod_{r=1}^{|t_i|} P(c_r|c_{r-n_{max}+1}^{r-1}, l_i) \quad (2)$$

In Equation (2), $t_i = c_1^{|t_i|}$ and $|t_i|$ is the length of t_i with respect to the key entry sequence. For example, in the case of t_i="ikura", $|t_i| = 5$ and c_1="i", c_2="k", c_3="u" and $|t_i|$=5. Here, each probability $P(c_r|c_{r-n_{max}+1}^{r-1}, l_i)$ needs to be smoothed.

Values of $P(t_i|l_i)$ are estimated from monolingual corpora. If the corpora are large, $P(t_i|l_i)$ is estimated from the token frequencies. However, because large corpora are not always available, especially for minor languages, $P(t_i|l_i)$ is estimated using smoothed character-based n-grams. *Prediction by Partial Matching*, or PPM is adopted for this task, since it naturally incorporates online learning and it is effective in various NLP tasks as reported in (Teahan et al., 2000) and (Tanaka-Ishii, 2006). PPM uses $c_1^{r-n_{max}}$ as a corpus for training. PPM is designed to predict the next c_r by estimating the n_{max}-gram probability $P(c_r|c_{r-n_{max}+1}^{r-1})$ using backing-off techniques with regard to the current context. Precisely, the probability is estimated as a weighted sum of different $(n+1)$-gram probabilities up to a fixed n_{max}-gram as follows:

$$P(c_r|c_{r-n_{max}+1}^{r-1}) = \sum_{n=-1}^{n_{max}-1} w_n p_n(c_r) \quad (3)$$

The weights w_n are determined through *escape probabilities*. Depending on how the escape probabilities are calculated, there are several PPM variants, which are named PPMA, PPMB, PPMC, and so on. PPMC, the one that we have used, is also known as Witten-Bell smoothing in the NLP field (Manning and Schuetze, 1999). The escape probabilities are defined as follows.

$$w_n = (1 - e_n) \prod_{n'=n+1}^{n_{cont}} e_{n'} \quad (-1 \leq n < n_{cont}) \quad (4)$$

$$w_{n_{cont}} = (1 - e_n)$$

Here, n_{cont} is defined as the maximum n that satisfies $X_n \neq 0$. Let X_n be the number of c_{r-n}^{r-1}, x_n be the number of c_{r-n}^{r} and q_n be the number of different keycodes followed by c_{r-n}^{r-1} found in c_1^{r-n-1}.

Using these notations, $p_n(c_r)$ is defined as

$$p_n(c_r) = \frac{x_n}{X_n} \quad (5)$$

In PPMC, the escape probabilities are calculated as

$$e_n = \frac{q_n}{X_n + q_n} \quad (6)$$

For further details, see (Bell et al., 1990).

3.3 Language Transition Probabilities

Only a small corpus is typically available to estimate $P(l_m|l_{m-k_{max}+1}^{m-1})$, where k_{max} is the longest k-gram in the language sequence to be considered. Thus, the transition probability is estimated on-line, making use of language that will be corrected interactively by the user. For this on-line learning, we adopted PPM as well as the output probabilities.

Note that a large k_{max} may reduce accuracy, which is intuitively explained as follows. While there is typically a high probability that the subsequent language will be the same as the current language, it is unlikely that any language sequence will have long regular patterns. Therefore, k_{max} should be fixed according to this consideration.

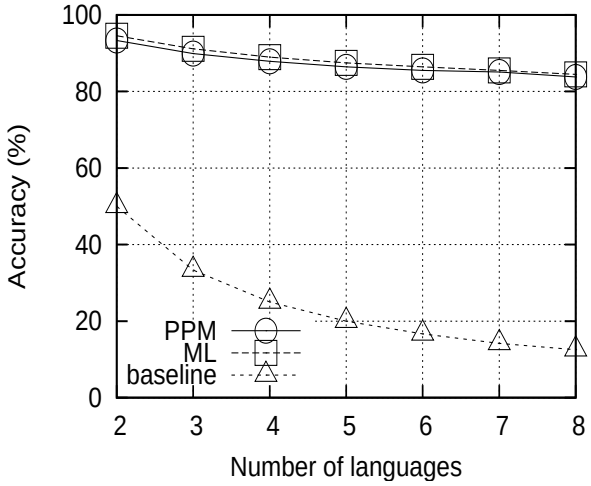


Figure 5: Detection Accuracy Test1

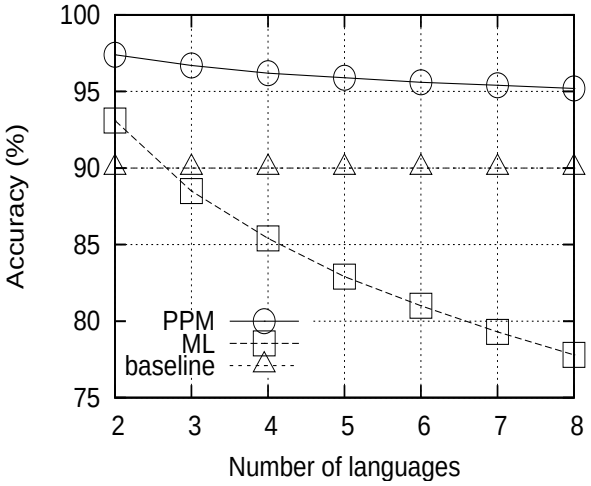


Figure 6: Detection Accuracy Test2

4 Evaluation

We evaluated TypeAny with respect to two measures: language detection capability when using artificially generated multilingual corpora, and the number of required control actions when using actual multilingual corpora.

4.1 Language Detection Accuracy

The ideal experiment would be to use actual multilingual corpora for many language sets. However, it is still difficult to collect a large amount of multilingual corpora with adequate quality for the test data of languages.

Therefore, we measured the language detection accuracies using artificially generated multilingual corpora by mixing monolingual corpora for every combination varying from two to eight languages.

First, the following monolingual corpora were collected: editions of the Mainichi newspaper in 2004 for Japanese, the Peking University corpus for Chinese, and the Leipzig corpora (Biemann et al., 2007) for English, French, German, Estonian, Finnish and Turkish. The text of each of these corpora was transformed into a sequence of key entries.

Two test sets, Test1 and Test2, were generated by using different mixture rates. In Test1, languages were mixed uniformly and randomly, whereas in Test2 a major language accounted for 90% of the text and the remaining 10% included different languages chosen uniformly and randomly. Test2 is more realistic since a document is usually composed in one major language.

The output and language transition probabilities were estimated and smoothed using PPMC as described in §3. Since part of the target of the experiment was to clarify the relation between learning size and accuracy, the output probabilities and transition probabilities were *not* trained on-line while the text was entered using PPMC, thus accuracy was measured by fixing the language model at this initial state. We used $n_{max} = 5$ for the output probability and $k_{max} = 1$ for the transition probability since the distribution of languages in the corpus was uniform here as we generated it uniformly. (See formula (4) in §3.2).

A 10-fold cross validation was applied to the generated corpora. Each generated corpus was 111 Kbytes in size, consisting of a disjoint 100-Kbyte training part and an 11-Kbyte testing part. The output probabilities were trained using the 100-Kbyte training part. The language transition probabilities were trained using about 2000 tokens.

The results for Test1 and Test2 are shown in Figure 5 and Figure 6, respectively. The horizontal axis shows the number of languages and the vertical axis shows the detection accuracy. There are three lines: PPM indicates that the transition probabilities were trained by PPM; ML indicates that no transition probability was used and the language was detected using only output probabilities (maximum likelihood only); Baseline is the accuracy when the most frequent language is always selected.

As shown in Figure 5 (Test1), when the mixture was uniform, the PPM performance was slightly lower but very close to that of ML. This was because PPM would be theoretically equivalent to ML with infinite learning of language transition probabilities, since languages were uniformly distributed in Test1. These results show that our PPM for transition probabilities learns this uniformity in Test1.

As shown in Figure 6, PPM clearly outperformed ML in Test2. This was because ML has no way to learn the transition probability, which was biased with the major language being used 90% of the time. This shows that the introduction of language transition probabilities accounts for higher performance. Interestingly, ML falls below the baseline case when more than three languages were used in Test2, a situation that has rarely been considered in previous studies. This suggests that language detection using only ML requires large corpora for learning to select one appropriate language, and that this requirement can be alleviated by using PPM.

Another finding is that the detection accuracy depends on the language set. For example, the accuracy for language sets consisting of both French and English tended to be lower than for other language sets due to the spelling closeness between these two languages. For example, the accuracy for test data consisting of 90% English, 5% French and 5% German was 94.4%. This is not surprising since the detection was made only *within* a token (which corresponds to a word in European languages): naturally there were many words whose language was ambiguous within the test set. In contrast, high accuracies were obtained for test sets consisting of languages more different in their nature. We obtained 97.5% accuracy for test data consisting of 90% English, 5% Finnish and 5% Turkish; this accuracy was higher than the average for all test sets.

4.2 Number of Control Actions

The second evaluation was done to compare the number of control actions needed to switch languages with TypeAny and with a conventional method. As mentioned in §1, three types of *keyboard actions* are used when entering text. Our work only concerns the control action to switch language, though, and the comparison in this section focuses on this type of action.

Table 1: Articles Used in the Decrease Test

article	Article 1	Article 2
Foreign tokens	286	55
Total tokens	1725	5100
Inclusion ratio	16.6%	1.1%
languages	E., J.	E., J., C.[7]
content	Introduction of Japanese phrases for traveling	About tofu (bean curd)
Source	Wikitravel [8]	Wikipedia [9]

Table 2: Required Number of Control Actions

		Article 1	Article 2
Conventional Number of switches		572 (100%)	110 (100%)
Ours	**Failure A**	2.8%	3.6%
	Failure B	1.6%	2.7%
Total Failures		4.4%	6.3%
Decrease		95.6%	93.6%

This evaluation was done using two samples of actual multilingual text collected from the Web. The features of these samples are shown in the top block of Table 1. In both cases, the major language was English.

For each of these articles, the number of control actions required with the conventional method and with TypeAny was measured. The conventional method requires a control action every time the language switches. For TypeAny, control actions are required only to correct language detection failures. In both cases, the action required to switch languages or correct the language was counted as one action.

For the language model, the output probabilities were first trained using the 100-Kbyte monolingual corpora collected for the previous evaluation. The transition probabilities were not trained beforehand; i.e., the system initially regarded the languages to be uniformly distributed. Since this experiment was intended to simulate a realistic case, both output and

[7] E.: English, J.: Japanese and C.: Chinese.
[8] http://en.wikitravel.org/
[9] http://en.wikipedia.org/

transition probabilities were trained on-line using PPMC while the text was entered. Here, both n_{max} and k_{max} were set at 5.

The results are shown in Table 2. First, some detection errors occurred for Article 2 because "tofu" was detected as Japanese at the beginning of entry, even though it was used as an English word in the original text. As noted at the end of §2, such loan words can cause errors. However, since our system uses PPM and learns on-line, our system learned that "tofu" had to be English, and such detection errors occurred *only* at the beginning of the text.

Consequently, there was a substantial decrease in the number of necessary control actions with TypeAny, over 93%, for both articles. An especially large decrease was observed for Article 2, even though the text was almost all in English (98.9%). There was only a small increase in the incidence rate of **Failure B** for Article 2, so the total decrease in the number of required actions was still large, putting to rest the concern discussed in §2. These results demonstrate the effectiveness of our approach.

5 Conclusion

TypeAny is a novel multilingual text input interface in which the languages used for entries are detected automatically. We modeled the language detection as an HMM process whose transition probabilities are estimated by on-line learning through the PPM method.

This system achieved language detection accuracy of 96.7% in an evaluation where it had to choose the appropriate language from among three languages with the major language accounting for 90% of the sample. In addition, the number of control actions required to switch IMEs was decreased by over 93%. These results show the promise of our system and suggest that it will work well under realistic circumstances.

An interesting objection might be raised to the conclusions of this study: some users might find it difficult to watch the locale window all the time and prefer the conventional method despite having to work with a large number of key types. We plan to examine and clarify the cognitive load of such users in our future work.

References

B. Alex, A. Dubey, and F. Keller. 2007. Using foreign inclusion detection to improve parsing performance. In *Proceedings of EMNLP-CoNLL*, Prague, Czech, June.

B. Alex. 2005. An unsupervised system for identifying English inclusions in German text. In *Proceedings of the ACL Student Research Workshop*, pages 133–138, Ann Arbor, Michigan, June. Association for Computational Linguistics.

T. C. Bell, J. G. Clear, and I. H. Witten. 1990. *Text Compression*. Prentice-Hall, New Jersey.

C. Biemann, G. Heyer, U. Quasthoff, and M. Richter. 2007. The Leipzig corpora collection - monolingual corpora of standard size. In *Proceedings of Corpus Linguistics*, Birmingham, United Kingdom, July.

Z. Chen and K. Lee. 2000. A new statistical approach to Chinese input. In *The 38th Annual Meeting of the Association for Computer Linguistics*, pages 241–247, Hong Kong, October.

I. S. MacKenzie and K. Tanaka-Ishii. 2007. *Text Entry Systems —Mobility, Accessibility, Universality–*. Morgan Kaufmann.

C. D. Manning and H. Schuetze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

J.-C. Marcadet, V. Fischer, and C. Waast-Richard. 2005. A transformation-based learning approach to language identification for mixed-lingual text-to-speech synthesis. In *Interspeech 2005 - ICSLP*, pages 2249–2252, Lisbon, Portugal.

K. N. Murthy and G. B. Kumar. 2006. Language identification from small text samples. *Journal of Quantitative Linguistics*, 13:57–80.

B. Pfister and H. Romsdorfer. 2003. Mixed-lingual analysis for polyglot TTS synthesis. In *Eurospeech*, pages 2037–2040, Geneva, Switzerland.

K. Tanaka-Ishii. 2006. Word-based text entry techniques using adaptive language models. *Journal of Natural Language Engineering*, 13(1):51–74.

W. J. Teahan, Y. Wen, R. MacNab, and I. H. Witten. 2000. A compression-based algorithm for Chinese word segmentation. In *Computational Linguistics*, volume 26, pages 375–393.

Using Contextual Speller Techniques and Language Modeling for ESL Error Correction

Michael Gamon*, Jianfeng Gao*, Chris Brockett*, Alexandre Klementiev[+], William B. Dolan*, Dmitriy Belenko*, Lucy Vanderwende*

*Microsoft Research
One Microsoft Way
Redmond, WA 98052
{mgamon,jfgao,chrisbkt,billdol,
dmitryb,lucyv}@microsoft.com

[+]Dept. of Computer Science
University of Illinois
Urbana, IL 61801
klementi@uiuc.edu

Abstract

We present a modular system for detection and correction of errors made by non-native (English as a Second Language = ESL) writers. We focus on two error types: the incorrect use of determiners and the choice of prepositions. We use a decision-tree approach inspired by contextual spelling systems for detection and correction suggestions, and a large language model trained on the Gigaword corpus to provide additional information to filter out spurious suggestions. We show how this system performs on a corpus of non-native English text and discuss strategies for future enhancements.

1 Introduction

English is today the de facto lingua franca for commerce around the globe. It has been estimated that about 750M people use English as a second language, as opposed to 375M native English speakers (Crystal 1997), while as much as 74% of writing in English is done by non-native speakers. However, the errors typically targeted by commercial proofing tools represent only a subset of errors that a non-native speaker might make. For example, while many non-native speakers may encounter difficulty choosing among prepositions, this is typically not a significant problem for native speakers and hence remains unaddressed in proofing tools such as the grammar checker in Microsoft Word (Heidorn 2000). Plainly there is an opening here for automated proofing tools that are better geared to the non-native users.

One challenge that automated proofing tools face is that writing errors often present a semantic dimension that renders it difficult if not impossible to provide a single correct suggestion. The choice of definite versus indefinite determiner—a common error type among writers with a Japanese, Chinese or Korean language background owing to the lack of overt markers for definiteness and indefiniteness—is highly dependent on larger textual context and world knowledge. It seems desirable, then, that proofing tools targeting such errors be able to offer a range of plausible suggestions, enhanced by presenting real-world examples that are intended to inform a user's selection of the most appropriate wording in the context[1].

2 Targeted Error Types

Our system currently targets eight different error types:

1. Preposition presence and choice:
   In the other hand, ... (On the other hand ...)
2. Definite and indefinite determiner presence and choice:
   I am teacher... (am a teacher)
3. Gerund/infinitive confusion:
   I am interesting in this book. (interested in)
4. Auxiliary verb presence and choice:
   My teacher does is a good teacher (my teacher is...)

[1] Liu et al. 2000 take a similar approach, retrieving example sentences from a large corpus.

5. Over-regularized verb inflection:
   I writed a letter (wrote)
6. Adjective/noun confusion:
   This is a China book (Chinese book)
7. Word order (adjective sequences and nominal compounds):
   I am a student of university (university student)
8. Noun pluralization:
   They have many knowledges (much knowledge)

In this paper we will focus on the two most prominent and difficult errors: choice of determiner and prepositions. Empirical justification for targeting these errors comes from inspection of several corpora of non-native writing. In the NICT Japanese Learners of English (JLE) corpus (Izumi et al. 2004), 26.6% of all errors are determiner related, and about 10% are preposition related, making these two error types the dominant ones in the corpus. Although the JLE corpus is based on transcripts of spoken language, we have no reason to believe that the situation in written English is substantially different. The Chinese Learners of English Corpus (CLEC, Gui and Yang 2003) has a coarser and somewhat inconsistent error tagging scheme that makes it harder to isolate the two errors, but of the non-orthographic errors, more than 10% are determiner and number related. Roughly 2% of errors in the corpus are tagged as preposition-related, but other preposition errors are subsumed under the "collocation error" category which makes up about 5% of errors.

3 Related Work

Models for determiner and preposition selection have mostly been investigated in the context of sentence realization and machine translation (Knight and Chander 1994, Gamon et al. 2002, Bond 2005, Suzuki and Toutanova 2006, Toutanova and Suzuki 2007). Such approaches typically rely on the fact that preposition or determiner choice is made in otherwise native-like sentences. Turner and Charniak (2007), for example, utilize a language model based on a statistical parser for Penn Tree Bank data. Similarly, De Felice and Pulman (2007) utilize a set of sophisticated syntactic and semantic analysis features to predict 5 common English prepositions. Obviously, this is impractical in a setting where noisy non-native text is subjected to proofing. Meanwhile, work on automated error detection on non-native text focuses primarily on detection of errors, rather than on the more difficult task of supplying viable corrections (e.g., Chodorow and Leacock, 2000). More recently, Han et al. (2004, 2006) use a maximum entropy classifier to propose article corrections in TESOL essays, while Izumi et al. (2003) and Chodorow et al. (2007) present techniques of automatic preposition choice modeling. These more recent efforts, nevertheless, do not attempt to integrate their methods into a more general proofing application designed to assist non-native speakers when writing English. Finally, Yi et al. (2008) designed a system that uses web counts to determine correct article usage for a given sentence, targeting ESL users.

4 System Description

Our system consists of three major components:

1. Suggestion Provider (SP)
2. Language Model (LM)
3. Example Provider (EP)

The Suggestion Provider contains modules for each error type discussed in section 2. Sentences are tokenized and part-of-speech tagged before they are presented to these modules. Each module determines parts of the sentence that may contain an error of a specific type and one or more possible corrections. Four of the eight error-specific modules mentioned in section 2 employ machine learned (classification) techniques, the other four are based on heuristics. Gerund/infinitive confusion and auxiliary presence/choice each use a single classifier. Preposition and determiner modules each use two classifiers, one to determine whether a preposition/article should be present, and one for the choice of preposition/article.

All suggestions from the Suggestion Provider are collected and passed through the Language Model. As a first step, a suggested correction has to have a higher language model score than the original sentence in order to be a candidate for being surfaced to the user. A second set of heuristic thresholds is based on a linear combination of class probability as assigned by the classifier and language model score.

The Example Provider queries the web for exemplary sentences that contain the suggested correction. The user can choose to consult this information to make an informed decision about the correction.

4.1 Suggestion Provider Modules for Determiners and Prepositions

The SP modules for determiner and preposition choice are machine learned components. Ideally, one would train such modules on large data sets of annotated errors and corrected counterparts. Such a data set, however, is not currently available. As a substitute, we are using native English text for training, currently we train on the full text of the English Encarta encyclopedia (560k sentences) and a random set of 1M sentences from a Reuters news data set. The strategy behind these modules is similar to a contextual speller as described, for example, in (Golding and Roth 1999). For each potential insertion point of a determiner or preposition we extract context features within a window of six tokens to the right and to the left. For each token within the window we extract its relative position, the token string, and its part-of-speech tag. Potential insertion sites are determined heuristically from the sequence of POS tags. Based on these features, we train a classifier for preposition choice and determiner choice. Currently we train decision tree classifiers with the WinMine toolkit (Chickering 2002). We also experimented with linear SVMs, but decision trees performed better overall and training and parameter optimization were considerably more efficient. Before training the classifiers, we perform feature ablation by imposing a count cutoff of 10, and by limiting the number of features to the top 75K features in terms of log likelihood ratio (Dunning 1993).

We train two separate classifiers for both determiners and preposition:

- decision whether or not a determiner/preposition should be present (*presence/absence* or ***pa classifier***)
- decision which determiner/preposition is the most likely choice, given that a determiner/preposition is present (*choice* or ***ch classifier***)

In the case of determiners, class values for the ch classifier are *a/an* and *the*. Preposition choice (equivalent to the "confusion set" of a contextual speller) is limited to a set of 13 prepositions that figure prominently in the errors observed in the JLE corpus: *about, as, at, by, for, from, in, like, of, on, since, to, with, than, "other"* (for prepositions not in the list).

The decision tree classifiers produce probability distributions over class values at their leaf nodes. For a given leaf node, the most likely preposition/determiner is chosen as a suggestion. If there are other class values with probabilities above heuristically determined thresholds[2], those are also included in the list of possible suggestions. Consider the following example of an article-related error:

I am teacher from Korea.

As explained above, the suggestion provider module for article errors consists of two classifiers, one for presence/absence of an article, the other for article choice. The string above is first tokenized and then part-of-speech tagged:

0/I/PRP 1/am/VBP 2/teacher/NN 3/from/IN 4/Korea/NNP 5/./.

Based on the sequence of POS tags and capitalization of the nouns, a heuristic determines that there is one potential noun phrase that could contain an article: *teacher*. For this possible article position, the article presence/absence classifier determines the probability of the presence of an article, based on a feature vector of pos tags and surrounding lexical items:

p(article + *teacher*) = 0.54

Given that the probability of an article in this position is higher than the probability of not having an article, the second classifier is consulted to provide the most likely choice of article:

p(*the*) = 0.04
p(*a/an*) = 0.96

Given this probability distribution, a correction suggestion *I am teacher from Korea -> I am **a** teacher from Korea* is generated and passed on to evaluation by the language model component.

4.2 The Language Model

The language model is a 5-gram model trained on the English Gigaword corpus (LDC2005T12). In order to preserve (singleton) context information as much as possible, we used interpolated Kneser-Ney smoothing (Kneser and Ney 1995) without count cutoff. With a 120K-word vocabulary, the trained language model contains 54 million bigrams, 338 million trigrams, 801 million 4-grams

[2] Again, we are working on learning these thresholds empirically from data.

and 12 billion 5-grams. In the example from the previous section, the two alternative strings of the original user input and the suggested correction are scored by the language model:

I am teacher from Korea. score = 0.19
*I am **a** teacher from Korea.* score = 0.60

The score for the suggested correction is significantly higher than the score for the original, so the suggested correction is provided to the user.

4.3 The Example Provider

In many cases, the SP will produce several alternative suggestions, from which the user may be able to pick the appropriate correction reliably. In other cases, however, it may not be clear which suggestion is most appropriate. In this event, the user can choose to activate the Example Provider (EP) which will then perform a web search to retrieve relevant example sentences illustrating the suggested correction. For each suggestion, we create an exact string query including a small window of context to the left and to the right of the suggested correction. The query is issued to a search engine, and the retrieved results are separated into sentences. Those sentences that contain the string query are added to a list of example candidates. The candidates are then ranked by two initially implemented criteria: Sentence length (shorter examples are preferred in order to reduce cognitive load) and context overlap (sentences that contain additional words from the user input are preferred). We have not yet performed a user study to evaluate the usefulness of the examples provided by the system. Some examples of usage that we retrieve are given below with the query string in boldface:

Original: *I am teacher from Korea.*
Suggestion: *I am a teacher from Korea.*
All top 3 examples: *I **am a teacher**.*

Original: *So Smokers have to see doctor more often than non-smokers.*
Suggestion: *So Smokers have to see a doctor more often than non-smokers.*
Top 3 examples:

1. *Do people going through withdrawal have to **see a doctor**?*
2. *Usually, a couple should wait to **see a doctor** until after they've tried to get pregnant for a year.*
3. *If you have had congestion for over a week, you should **see a doctor**.*

Original: *I want to travel Disneyland in March.*
Suggestion: *I want to travel **to** Disneyland in March.*
Top 3 examples:

1. *Timothy's wish was **to travel to Disneyland** in California.*
2. *Should you **travel to Disneyland** in California or to Disney World in Florida?*
3. *The tourists who **travel to Disneyland** in California can either choose to stay in Disney resorts or in the hotel for Disneyland vacations.*

5 Evaluation

We perform two different types of evaluation on our system. Automatic evaluation is performed on native text, under the assumption that the native text does not contain any errors of the type targeted by our system. For example, the original choice of preposition made in the native text would serve as supervision for the evaluation of the preposition module. Human evaluation is performed on non-native text, with a human rater assessing each suggestion provided by the system.

5.1 Individual SP Modules

For evaluation, we split the original training data discussed in section 4.1 into training and test sets (70%/30%). We then retrained the classifiers on this reduced training set and applied them to the held-out test set. Since there are two models, one for preposition/determiner presence and absence (*pa*), and one for preposition/determiner choice (*ch*), we report combined accuracy numbers of the two classifiers. *Votes(a)* stands for the counts of votes for class value = *absence* from *pa*, *votes(p)* stands for counts of votes for *presence* from *pa*. *Acc(pa)* is the accuracy of the *pa* classifier, *acc(ch)* the accuracy of the choice classifier. Combined accuracy is defined as in Equation 1.

$$\frac{acc(pa) * votes(a) + acc(ch) * acc(pa) * votes(p)}{total\ cases}$$

Equation 1: Combined accuracy of the presence/absence and choice models

The total number of cases in the test set is 1,578,342 for article correction and 1,828,438 for preposition correction.

5.1.1 Determiner choice

Accuracy of the determiner pa and ch models and their combination is shown in Table 1.

Model	pa	ch	combined
Accuracy	89.61%	85.97%	86.07%

Table 1: Accuracy of the determiner pa, ch, and combined models.

The baseline is 69.9% (choosing the most frequent class label *none*). The overall accuracy of this module is state-of-the-art compared with results reported in the literature (Knight and Chander 1994, Minnen et al. 2000, Lee 2004, Turner and Charniak 2007). Turner and Charniak 2007 obtained the best reported accuracy to date of 86.74%, using a Charniak language model (Charniak 2001) based on a full statistical parser on the Penn Tree Bank. These numbers are, of course, not directly comparable, given the different corpora. On the other hand, the distribution of determiners is similar in the PTB (as reported in Minnen et al. 2000) and in our data (Table 2).

	PTB	Reuters/Encarta mix
no determiner	70.0%	69.9%
the	20.6%	22.2%
a/an	9.4%	7.8%

Table 2: distribution of determiners in the Penn Tree Bank and in our Reuters/Encarta data.

Precision and recall numbers for both models on our test set are shown in Table 3 and Table 4.

Article pa classifier	precision	recall
presence	84.99%	79.54%
absence	91.43%	93.95%

Table 3: precision and recall of the article pa classifier.

Article ch classifier	precision	Recall
the	88.73%	92.81%
a/an	76.55%	66.58%

Table 4: precision and recall of the article ch classifier.

5.1.2 Preposition choice

The preposition choice model and the combined model achieve lower accuracy than the corresponding determiner models, a result that can be expected given the larger choice of candidates and hardness of the task. Accuracy numbers are presented in Table 5.

Model	pa	ch	combined
Accuracy	91.06%%	62.32%	86.07%

Table 5:Accuracy of the preposition pa, ch, and combined models.

The baseline in this task is 28.94% (using no preposition). Precision and recall numbers are shown in Table 6 and Table 7. From Table 7 it is evident that prepositions show a wide range of predictability. Prepositions such as *than* and *about* show high recall and precision, due to the lexical and morphosyntactic regularities that govern their distribution. At the low end, the semantically more independent prepositions *since* and *at* show much lower precision and recall numbers.

Preposition pa classifier	precision	recall
presence	90.82%	87.20%
absence	91.22%	93.78%

Table 6: Precision and recall of the preposition pa classifier.

Preposition ch classifier	precision	recall
other	53.75%	54.41%
in	55.93%	62.93%
for	56.18%	38.76%
of	68.09%	85.85%
on	46.94%	24.47%
to	79.54%	51.72%
with	64.86%	25.00%
at	50.00%	29.67%
by	42.86%	60.46%
as	76.78%	64.18%
from	81.13%	39.09%
since	50.00%	10.00%
about	93.88%	69.70%
than	95.24%	90.91%

Table 7: Precision and recall of the preposition ch classifier.

Chodorow et al. (2007) present numbers on an independently developed system for detection of preposition error in non-native English. Their approach is similar to ours in that they use a classifier with contextual feature vectors. The major differences between the two systems are the additional use of a language model in our system and, from a usability perspective, in the example provider module we added to the correction process. Since both systems are evaluated on different data sets[3], however, the numbers are not directly comparable.

5.2 Language model Impact

The language model gives us an additional piece of information to make a decision as to whether a correction is indeed valid. Initially, we used the language model as a simple filter: any correction that received a lower language model score than the original was filtered out. As a first approximation, this was an effective step: it reduced the number of preposition corrections by 66.8% and the determiner corrections by 50.7%, and increased precision dramatically. The language model alone, however, does not provide sufficient evidence: if we produce a full set of preposition suggestions for each potential preposition location and rank these suggestions by LM score alone, we only achieve 58.36% accuracy on Reuters data.

Given that we have multiple pieces of information for a correction candidate, namely the class probability assigned by the classifier and the language model score, it is more effective to combine these into a single score and impose a tunable threshold on the score to maximize precision. Currently, this threshold is manually set by analyzing the flags in a development set.

5.3 Human Evaluation

A complete human evaluation of our system would have to include a thorough user study and would need to assess a variety of criteria, from the accuracy of individual error detection and corrections to the general helpfulness of real web-based example sentences. For a first human evaluation of our system prototype, we decided to simply address the question of accuracy on the determiner and preposition choice tasks on a sample of non-native text.

For this purpose we ran the system over a random sample of sentences from the CLEC corpus (8k for the preposition evaluation and 6k for the determiner evaluation). An independent judge annotated each flag produced by the system as belonging to one of the following categories:

- (1) the correction is valid and fixes the problem
- (2) the error is correctly identified, but the suggested correction does not fix it
- (3) the original and the rewrite are both equally good
- (4) the error is at or near the suggested correction, but it is a different kind of error (not having to do with prepositions/determiners)
- (5) There is a spelling error at or near the correction
- (6) the correction is wrong, the original is correct

Table 8 shows the results of this human assessment for articles and prepositions.

	Articles (6k sentences)		**Prepositions (8k sentences)**	
	count	**ratio**	**count**	**ratio**
(1) correction is valid	240	55%	165	46%
(2) error identified, suggestion does not fix it	10	2%	17	5%
(3) original and suggestion equally good	17	4%	38	10%
(4) misdiagnosis	65	15%	46	13%
(5) spelling error near correction	37	8%	20	6%
(6) original correct	70	16%	76	21%

Table 8: Article and preposition correction accuracy on CLEC data.

The distribution of corrections across deletion, insertion and substitution operations is illustrated in Table 9. The most common article correction is insertion of a missing article. For prepositions, substitution is the most common correction, again an expected result given that the presence of a

[3] Chodorow et al. (2007) evaluate their system on proprietary student essays from non-native students, where they achieve 77.8% precision at 30.4% recall for the preposition substitution task.

preposition is easier to determine for a non-native speaker than the actual choice of the correct preposition.

	deletion	insertion	substitution
Articles	8%	79%	13%
Prepositions	15%	10%	76%

Table 9: Ratio of deletion, insertion and substitution operations.

6 Conclusion and Future Work

Helping a non-native writer of English with the correct choice of prepositions and definite/indefinite determiners is a difficult challenge. By combining contextual speller based methods with language model scoring and providing web-based examples, we can leverage the combination of evidence from multiple sources.

The human evaluation numbers presented in the previous section are encouraging. Article and preposition errors present the greatest difficulty for many learners as well as machines, but can nevertheless be corrected even in extremely noisy text with reasonable accuracy. Providing contextually appropriate real-life examples alongside with the suggested correction will, we believe, help the non-native user reach a more informed decision than just presenting a correction without additional evidence and information.

The greatest challenge we are facing is the reduction of "false flags", i.e. flags where both error detection and suggested correction are incorrect. Such flags—especially for a non-native speaker—can be confusing, despite the fact that the impact is mitigated by the set of examples which may clarify the picture somewhat and help the users determine that they are dealing with an inappropriate correction. In the current system we use a set of carefully crafted heuristic thresholds that are geared towards minimizing false flags on a development set, based on detailed error analysis. As with all manually imposed thresholding, this is both a laborious and brittle process where each retraining of a model requires a re-tuning of the heuristics. We are currently investigating a learned ranker that combines information from language model and classifiers, using web counts as a supervision signal.

7 Acknowledgements

We thank Claudia Leacock (Butler Hill Group) for her meticulous analysis of errors and human evaluation of the system output, as well as for much invaluable feedback and discussion.

References

Bond, Francis. 2005. *Translating the Untranslatable: A Solution to the Problem of Generating English Determiners.* CSLI Publications.

Charniak, Eugene. 2001. Immediate-head parsing for language models. In *Proceedingsof the 39th Annual Meeting of the Association for Computational Linguistics*, pp 116-123.

Chickering, David Maxwell. 2002. The WinMine Toolkit. Microsoft Technical Report 2002-103.

Chodorow, Martin, Joel R. Tetreault and Na-Rae Han. 2007. Detection of Grammatical Errors Involving Prepositions. In *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, pp 25-30.

Crystal, David. 1997. *Global English.* Cambridge University Press.

Rachele De Felice and Stephen G Pulman. 2007. *Automatically acquiring models of preposition use.* Proceedings of the ACL-07 Workshop on Prepositions.

Dunning, Ted. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19:61-74.

Gamon, Michael, Eric Ringger, and Simon Corston-Oliver. 2002. Amalgam: A machine-learned generation module. Microsoft Technical Report, MSR-TR-2002-57.

Golding, Andrew R. and Dan Roth. 1999. A Winnow Based Approach to Context-Sensitive Spelling Correction. *Machine Learning*, pp. 107-130.

Gui, Shicun and Huizhong Yang (eds.). 2003. *Zhongguo Xuexizhe Yingyu Yuliaohu*. (*Chinese Learner English Corpus*). Shanghai Waiyu Jiaoyu Chubanshe..

Han, Na-Rae., Chodorow, Martin and Claudia Leacock. 2004. Detecting errors in English article usage with a maximum entropy classifier trained on a large, diverse corpus. *Proceedings of the 4th international conference on language resources and evaluation*, Lisbon, Portugal.

Han, Na-Rae. Chodorow, Martin., and Claudia Leacock. (2006). Detecting errors in English article usage by non-native speakers. *Natural Language Engineering, 12*(2), 115-129.

Heidorn, George. 2000. Intelligent Writing Assistance. In Robert Dale, Herman Moisl, and Harold Somers (eds.). *Handbook of Natural Language Processing.* Marcel Dekker. pp 181 -207.

Izumi, Emi, Kiyotaka Uchimoto and Hitoshi Isahara. 2004. The NICT JLE Corpus: Exploiting the Language Learner's Speech Database for Research and Education. *International Journal of the Computer, the Internet and Management* 12:2, pp 119 -125.

Kneser, Reinhard. and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1. 1995. pp. 181–184.

Knight, Kevin and Ishwar Chander. 1994. Automatic Postediting of Documents. *Proceedings of the American Association of Artificial Intelligence,* pp 779-784.

Lee, John. 2004. Automatic Article Restoration. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 31-36.

Liu, Ting, Mingh Zhou, JianfengGao, Endong Xun, and Changning Huan. 2000. PENS: A Machine-Aided English Writing System for Chinese Users. *Proceedings of ACL 2000*, pp 529-536.

Minnen, Guido, Francis Bond and Ann Copestake. 2000. Memory-Based Learning for Article Generation. *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic* Workshop, pp 43-48.

Suzuki, Hisami and Kristina Toutanova. 2006. Learning to Predict Case Markers in Japanese. *Proceedings of* COLING-ACL, pp. 1049-1056.

Toutanova, Kristina and Hisami Suzuki. 2007 Generating Case Markers in Machine Translation. *Proceedings of NAACL-HLT*.

Turner, Jenine and Eugene Charniak. 2007. Language Modeling for Determiner Selection. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pp 177-180.

Yi, Xing, Jianfeng Gao and William B. Dolan. 2008. Web-Based English Proofing System for English as a Second Language Users. *To be presented at IJCNLP 2008.*

Bilingual Synonym Identification with Spelling Variations

Takashi Tsunakawa* **Jun'ichi Tsujii*†‡**
*Department of Computer Science,
Graduate School of Information Science and Technology, University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-0033 Japan
†School of Computer Science, University of Manchester
Oxford Road, Manchester, M13 9PL, UK
‡National Centre for Text Mining 131 Princess Street, Manchester, M1 7DN, UK
{tuna, tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper proposes a method for identifying synonymous relations in a bilingual lexicon, which is a set of translation-equivalent term pairs. We train a classifier for identifying those synonymous relations by using spelling variations as main clues. We compared two approaches: the direct identification of bilingual synonym pairs, and the merger of two monolingual synonyms. We showed that our approach achieves a high pair-wise precision and recall, and outperforms the baseline method.

1 Introduction

Automatically collecting synonyms from language resources is an ongoing task for natural language processing (NLP). Most NLP systems have difficulties in dealing with synonyms, which are different representations that have the same meaning in a language. Information retrieval (IR) could leverage synonyms to improve the coverage of search results (Qiu and Frei, 1993). For example, when we input the query 'transportation in India' into an IR system, the system can expand the query to its synonyms; e.g. 'transport' and 'railway', to find more documents.

This paper proposes a method for the automatic identification of bilingual synonyms in a bilingual lexicon, with spelling variation clues. A bilingual synonym set is a set of translation-equivalent term pairs sharing the same meaning. Although a number of studies have aimed at identifying synonyms, this is the first study that simultaneously finds synonyms in two languages, to our best knowledge.

Let us consider the case where a user enters the Japanese query '*kōjō*' (工場, industrial plant) into a cross-lingual IR system to find English documents. After translating the query into the English translation equivalent, 'plant,' the cross-lingual IR system may expand the query to its English synonyms, e.g. 'factory,' and 'workshop,' and retrieve documents that include the expanded terms. However, the term 'plant' is ambiguous; the system may also expand the query to 'vegetable,' and the system is prevented by the term which is different from our intention. In contrast, the system can easily reject the latter expansion, 'vegetable,' if we are aware of bilingual synonyms, which indicate synonymous relations over bilingual lexicons: (*kōjō*, plant) $\sim$ (*kōjō*, factory) and (*shokubutsu*[1], plant) $\sim$ (*shokubutsu*, vegetable)[2] (See Figure 1). The expression of the translation equivalent, (*kōjō*, plant), helps a cross-lingual IR system to retrieve documents that include the term 'plant,' used in the meaning for *kōjō*, or industrial plants.

We present a supervised machine learning approach for identifying bilingual synonyms. Designing features for bilingual synonyms such as spelling variations and bilingual associations, we train a classifier with a manually annotated bilingual lexicon with synonymous information. In order to evaluate the performance of our method, we carried out experiments to identify bilingual synonyms by two approaches: the direct identification of bilingual synonym pairs, and bilingual synonym pairs merged from two monolingual synonym lists. Experimental results show that our approach achieves the F-scores

[1] *Shokubutsu* (植物) means botanical plant.
[2] '$\sim$' represents the synonymous relation.

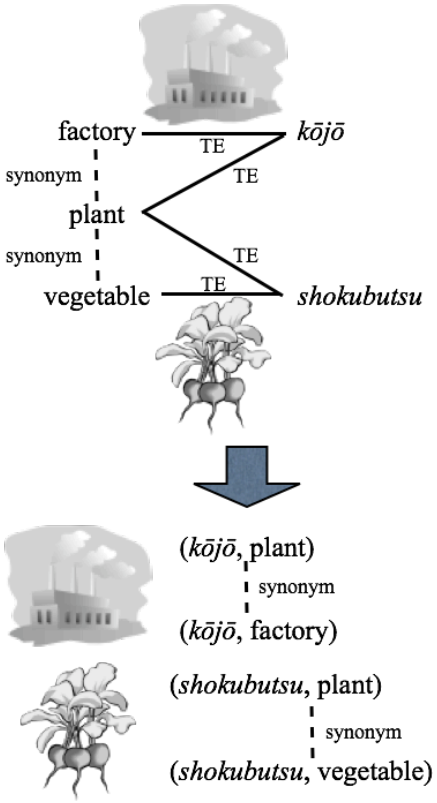


Figure 1: An example of an ambiguous term 'plant', and the synonyms and translation equivalents (TE)

89.3% in the former approach and 91.4% in the latter, thus outperforming the baseline method that employs only bilingual relations as its clues.

The remainder of this paper is organized as follows. The next section describes related work on synonym extraction and spelling variations. Section 3 describes the overview and definition of bilingual synonyms, the proposed method and employed features. In Section 4 we evaluate our method and conclude this paper.

2 Related work

There have been many approaches for detecting synonyms and constructing thesauri. Two main resources for synonym extraction are large text corpora and dictionaries.

Many studies extract synonyms from large monolingual corpora by using context information around target terms (Croach and Yang, 1992; Park and Choi, 1996; Waterman, 1996; Curran, 2004). Some researchers (Hindle, 1990; Grefenstette, 1994; Lin, 1998) classify terms by similarities based on their distributional syntactic patterns. These methods often extract not only synonyms, but also semantically related terms, such as antonyms, hyponyms and coordinate terms such as 'cat' and 'dog.'

Some studies make use of bilingual corpora or dictionaries to find synonyms in a target language (Barzilay and McKeown, 2001; Shimohata and Sumita, 2002; Wu and Zhou, 2003; Lin et al., 2003). Lin et al. (2003) chose a set of synonym candidates for a term by using a bilingual dictionary and computing distributional similarities in the candidate set to extract synonyms. They adopt the bilingual information to exclude non-synonyms (e.g., antonyms and hyponyms) that may be used in the similar contexts. Although they make use of bilingual dictionaries, this study aims at finding bilingual synonyms directly.

In the approaches based on monolingual dictionaries, the similarities of definitions of lexical items are important clues for identifying synonyms (Blondel et al., 2004; Muller et al., 2006). For instance, Blondel et al. (2004) constructed an associated dictionary graph whose vertices are the terms, and whose edges from v_1 to v_2 represent occurrence of v_2 in the definition for v_1. They choose synonyms from the graph by collecting terms pointed to and from the same terms.

Another strategy for finding synonyms is to consider the terms themselves. We divide it into two approaches: rule-based and distance-based.

Rule-based approaches implement rules with language-specific patterns and detect variations by applying rules to terms. Stemming (Lovins, 1968; Porter, 1980) is one of the rule-based approaches, which cuts morphological suffix inflections, and obtains the stems of words. There are other types of variations for phrases; for example, insertion, deletion or substitution of words, and permutation of words such as 'view point' and 'point of view' are such variations (Daille et al., 1996).

Distance-based approaches model the similarity or dissimilarity measure between two terms to find similar terms. The edit distance (Levenshtein, 1966) is the most widely-used measure, based on the minimum number of operations of insertion, deletion, or substitution of characters for transforming one term into another. It can be efficiently calculated by using

Term pairs	Concept
p_1 = (*shōmei* (照明), light)	c_1
p_2 = (*shōmei*, lights)	c_1
p_3 = (*karui* (軽い), light)	c_2
p_4 = (*raito* (ライト), light)	c_1, c_2
p_5 = (*raito*, lights)	c_1
p_6 = (*raito*, right)	c_3
p_7 = (*migi* (右), right)	c_3
p_8 = (*raito*, right fielder)	c_4
p_9 = (*kenri* (権利), right)	c_5
p_{10} = (*kenri*, rights)	c_5

Table 1: An Example of a bilingual lexicon and synonym sets (concepts)

	J terms	E terms	Description
c_1	*shōmei, raito*	light, lights	illumination
c_2	*karui, raito*	light	lightweight
c_3	*migi, raito*	right	right-side
c_4	*raito*	right fielder	(baseball)
c_5	*kenri*	right, rights	privilege

Table 2: The concepts in Table 1

a dynamic programming algorithm, and we can set the costs/weights for each character type.

3 Bilingual Synonyms and Translation Equivalents

This section describes the notion of bilingual synonyms and our method for identifying the synonymous pairs of translation equivalents. We consider a bilingual synonym as a set of translation-equivalent term pairs referring to the same concept.

Tables 1 and 2 are an example of bilingual synonym sets. There are ten Japanese-English translation-equivalent term pairs and five bilingual synonym sets in this example. A Japanese term '*raito*' is the phonetic transcription of both 'light' and 'right,' and it covers four concepts described by the three English terms. Figure 2 illustrates the relationship among these terms. The synonymous relation and the translation equivalence are considered to be similar in that two terms share the meanings. Following synonymous relation between terms in one language, we deal with the synonymous relation between bilingual translation-equivalent term pairs

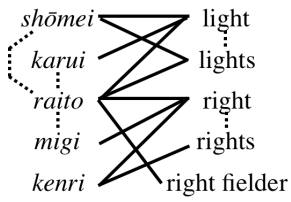


Figure 2: Relations among terms in Table 2
Solid lines show that two terms are translation equivalents, while dotted lines show that two terms are (monolingual) synonyms.

as bilingual synonyms. The advantage of managing the lexicon in the format of bilingual synonyms is that we can facilitate to tie the concepts and the terms.

3.1 Definitions

Let E and F be monolingual lexicons. We first assume that a term $e \in E$ (or $f \in F$) refers to one or more concepts, and define that a term e is a synonym[3] of $e'(\in E)$ if and only if e and e' share an identical concept[4]. Let '$\sim$' represent the synonymous relation, and this relation is not transitive because a term often has several concepts:

$$e \sim e' \ \wedge \ e' \sim e'' \not\Longrightarrow e \sim e''. \tag{1}$$

We define a synonym set (synset) E_c as a set whose elements share an identical concept c: $E_c = \{e \in E | \forall e \text{ refers to } c\}$. For a term set $E_c(\subseteq E)$,

$$\begin{aligned} & E_c \text{ is a synonym set (synset)} \\ \Longrightarrow \quad & \forall e, e' \in E_c \quad e \sim e' \end{aligned} \tag{2}$$

is true, but the converse is not necessarily true, because of the ambiguity of terms. Note that one term can belong to multiple synonym sets from the definition.

Let $D(\subseteq F \times E)$ be a bilingual lexicon defined as a set of term pairs (f, e) $(f \in F, e \in E)$ satisfying that f and e refer to an identical concept. We

[3]For distinguishing from bilingual synonyms, we often call the synonym a monolingual synonym.

[4]The definition of concepts, that is, the criteria of deciding whether two terms are synonymous or not, is beyond the focus of this paper. We do not assume that related terms such as hypernyms, hyponyms and coordinates are kinds of synonyms. In our experiments the criteria depend on manual annotation of synonym IDs in the training data.

call these pairs translation equivalents, which refer to concepts that both f and e refer to. We define that two bilingual lexical items p and $p'(\in D)$ are *bilingual synonyms* if and only if p and p' refer to an identical concept in common with the definition of (monolingual) synonyms. This relation is not transitive again, and if $e \sim e'$ and $f \sim f'$, it is not necessarily true that $p \sim p'$:

$$e \sim e' \ \wedge \ f \sim f' \not\Longrightarrow p \sim p' \tag{3}$$

because of the ambiguity of terms. Similarly, we can define a bilingual synonym set (synset) D_c as a set whose elements share an identical meaning c: $D_c = \{p \in D | \forall p \text{ refers to } c\}$. For a set of translation eqiuvalents D_c,

$$D_c \text{ is a bilingual synonym set (synset)} \Longrightarrow \forall p, p' \in D_c \quad p \sim p' \tag{4}$$

is true, but the converse is not necessarily true.

3.2 Identifying bilingual synonym pairs

In this section, we describe an algorithm to identify bilingual synonym pairs by using spelling variation clues. After identifying the pairs, we can construct bilingual synonym sets by assuming that the converse of the condition (4) is true, and finding sets of bilingual lexical items in which all paired items are bilingual synonyms. We can see this method as the complete-linkage clustering of translation-equivalent term pairs. We can adopt another option to construct them by assuming also that the bilingual synonymous relation has transitivity: $p \sim p' \wedge p' \sim p'' \Longrightarrow p \sim p''$, and this can be seen as simple-linkage clustering. This simplified method ignores the ambiguity of terms, and it may construct a bilingual synonym sets which includes many senses. In spite of the risk, it is effective to find large synonym sets in case the bilingual synonym pairs are not sufficiently detected. In this paper we focus only on identifying bilingual synonym pairs and evaluating the performance of the identification.

We employ a supervised machine learning technique with features related to spelling variations and so on. Figure 3 shows the framework for this method. At first we prepare a bilingual lexicon with synonymous information as training data, and generate a list consisting of all bilingual lexical item

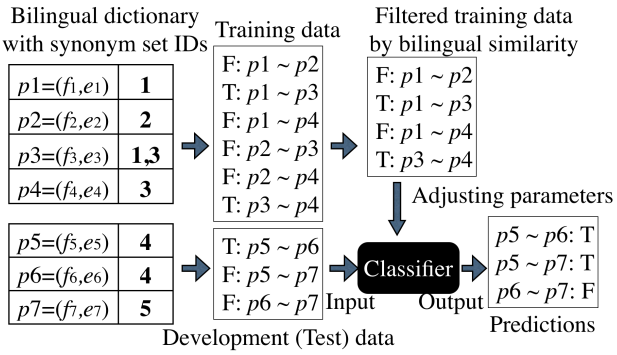


Figure 3: Overview of our framework

pairs in the bilingual lexicon. The presence or absence of bilingual synonymous relations is attached to each element of the list. Then, we build a classifier learned by training data, using a maximum entropy model (Berger et al., 1996) and the features related to spelling variations in Table 3.

We apply some preprocessings for extracting some features. For English, we transform all terms into lower-case, and do not apply any other transformations such as tokenization by symbols. For Japanese, we apply a morphological analyzer JUMAN (Kurohashi et al., 1994) and obtain *hiragana* representations[5] as much as possible[6]. We may require other language-specific preprocessings for applying this method to other languages.

We employed binary or real-valued features described in Table 3. Moreover, we introduce the following combinatorial features: $h_{1F} \wedge h_{1E}$, $\sqrt{h_{2F} \cdot h_{2E}}$, $\sqrt{h_{3F} \cdot h_{3E}}$, $h_{5E} \wedge h_{5F}$, $h_6 \cdot h_{2F}$ and $h_7 \cdot h_{2E}$.

3.2.1 Two approaches for identifying bilingual synonym pairs

There are two approaches for identifying bilingual synonym pairs: one is directly identifying whether two bilingual lexical items are bilingual synonyms ('bilingual' method), and another is first

[5] *Hiragana* is one of normalized representations of Japanese terms, which denotes how to pronounce the term. Japanese vocabulary has many of homonyms, which are semantically different but have the same pronunciation. Despite the risk of classifying homonyms into synonyms, we do not use original forms of Japanese terms because they are typically too short to extract character similarities.

[6] We keep unknown terms of JUMAN unchanged.

h_{1F}, h_{1E}: Agreement of the first characters	Whether the first characters match or not
h_{2F}, h_{2E}: Normalized edit distance	$1 - \frac{\mathrm{ED}(w,w')}{\max(\lvert w\rvert,\lvert w'\rvert)}$, where $\mathrm{ED}(w,w')$ is a non-weighted edit distance between w and w' and $\lvert w\rvert$ is the number of characters in w
h_{3F}, h_{3E}: Bigram similarity	$\frac{\lvert \mathrm{bigram}(w) \cap \mathrm{bigram}(w')\rvert}{\max(\lvert w\rvert,\lvert w'\rvert)-1}$, where $\mathrm{bigram}(w)$ is a multiset of character-based bigrams in w
h_{4F}, h_{4E}: Agreement or known synonymous relation of word sub-sequences	The count that sub-sequences of the target terms match as known terms or are in known synonymous relation
h_{5F}, h_{5E}: Existence of crossing bilingual lexical items	For bilingual lexical items (f_1, e_1) and (f_2, e_2), whether (f_1, e_2) (for h_{5F}) or (f_2, e_1) (for h_{5E}) is in the bilingual lexicon of the training set
h_6: Acronyms	Whether one English term is an acronym for another (Schwartz and Hearst, 2003)
h_7: *Katakana* variants	Whether one Japanese term is a *katakana* variant for another (Masuyama et al., 2004)

Table 3: Features used for identifying bilingual synonym pairs
h_{iF} is the feature value when the terms w and $w'(\in F)$ are compared in the i-th feature and so as h_{iE}. h_6 is only for English and h_7 is only for Japanese.

identifying monolingual synonyms in each language and then merging them according to the bilingual items ('monolingual' method). We implement these two approaches and compare the results. For identifying monolingual synonyms, we use features with bilingual items as follows: For a term pair e_1 and e_2, we obtain all the translation candidates $F_1 = \{f|(f, e_1) \in D\}$ and $F_2 = \{f'|(f', e_2) \in D\}$, and calculate feature values related to F_1 and/or F_2 by obtaining the maximum feature value using F_1 and/or F_2. After that, if all the following four conditions ($p_1 = (f_1, e_1) \in D$, $p_2 = (f_2, e_2) \in D$, $f_1 \sim e_1$ and $f_2 \sim e_2$) are satisfied, we assume that p_1 and p_2 are bilingual synonym pairs[7].

4 Experiment

4.1 Experimental settings

We performed experiments to identify bilingual synonym pairs by using the Japanese-English lexicon with synonymous information[8]. The lexicon consists of translation-equivalent term pairs extracted from titles and abstracts of scientific papers published in Japan. It contains many spelling variations and synonyms for constructing and maintaining the thesaurus of scientific terms and improving the coverage. Table 4 illustrates this lexicon.

	Total	train	dev.	test
$\lvert D\rvert$	210647	168837	20853	20957
$\lvert J\rvert$	136128	108325	13937	13866
$\lvert E\rvert$	115002	91057	11862	12803
Synsets	50710	40568	5071	5071
Pairs	814524	651727	77706	85091

Table 5: Statistics of the bilingual lexicon for our experiment
$|D|, |J|$, and $|E|$ are the number of bilingual lexical items, the number of Japanese vocabularies, and the number of English vocabularies, respectively. 'Synsets' and 'Pairs' are the numbers of synonym sets and synonym pairs, respectively.

Table 5 shows the statistics of the dictionary. We used information only synonym IDs and Japanese and English representations. We extract pairs of bilingual lexical items, and treat them as events for training of the maximum entropy method. The parameters were adjusted so that the performance is the best for the development set. For a monolingual method, we used $T_b = 0.8$, and for a bilingual method, we used $T_b = 0.7$.

4.2 Evaluation

We evaluated the performance of identifying bilingual synonym pairs by the pair-wise precision P,

[7] Actually, these conditions are not sufficient to derive the bilingual synonym pairs described in Section 3.1. We assume this approximation because there seems to be few counter examples in actual lexicons.

[8] This data was edited and provided by Japan Science and Technology Agency (JST).

Synset ID	J term	E term
130213	身体部位 (*shintai-bui*)	Body Regions
130213	身体部位 (*shintai-bui*)	body part
130213	身体部位 (*shintai-bui*)	body region
130213	身体部分 (*shintai-bubun*)	body part
130217	Douglas 窩 (Douglas-*ka*)	Douglas' Pouch
130217	Douglas か (Douglas-*ka*)	Douglas' Pouch
130217	ダグラス窩 (*Dagurasu-ka*)	pouch of Douglas
130217	ダグラスか (*Dagurasu-ka*)	pouch of Douglas
130217	直腸子宮窩 (*chokuchō-shikyū-ka*)	rectouterine pouch
130217	直腸子宮か (*chokuchō-shikyū-ka*)	rectouterine pouch

Table 4: A part of the lexicon used

Each bilingual synonym set consists of items that have the same synset ID. 部分 (*bubun*) is a synonym of 部位 (*bui*). か (*ka*) is a *hiragana* representation of 窩 (*ka*). ダグラス (*Dagurasu*) is a Japanese transcription of 'Douglas'.

recall R and F-score F defined as follows:

$$P = \frac{C}{T}, R = \frac{C}{N}, F = \frac{2PR}{P+R}, \quad (5)$$

where C, T and N are the number of correctly predicted pairs as synonyms, predicted pairs to become synonyms, and synonym pairs in the lexicon[9], respectively.

We compared the results with the baseline and the upper bound. The baseline assumes that each bilingual lexical item is a bilingual synonym if either the Japanese or English terms are identical. The upper bound assumes that all the monolingual synonyms are known and each bilingual item is a bilingual synonym if the Japanese terms and the English terms are synonymous. The baseline represents the performance when we do not consider spelling variations, and the upper bound shows the limitation of the monolingual approach.

4.3 Result

Table 6 shows the evaluation scores of our experiments. The 'monolingual' and 'bilingual' methods are described in Section 3.2.1. We obtained high precision and recall scores, although we used features primarily with spelling variations. Both methods significantly outperform the baseline, and show the importance of considering spelling variations.

[9] N includes the number of synonym pairs filtered out from training set by the bigram similarity threshold T_b.

Set	Method	Precision	Recall	F-score
dev.	baseline	0.977 (31845/32581)	0.410 (31845/77706)	0.577
	monolingual	**0.911** (74263/81501)	**0.956** (74263/77706)	**0.932**
	bilingual	0.879 (72782/82796)	0.937 (72782/77706)	0.907
	upper bound	0.984 (77706/78948)	1	0.992
test	baseline	0.972 (33382/34347)	0.392 (33382/85091)	0.559
	monolingual	**0.900** (79099/87901)	**0.930** (79099/85091)	**0.914**
	bilingual	0.875 (77640/88714)	0.912 (77640/85091)	0.893
	upper bound	0.979 (85091/86937)	1	0.989

Table 6: Evaluation scores

The 'monolingual' method achieved higher precision and recall than the 'bilingual' method. It indicates that monolingual synonym identification is effective in finding bilingual synonyms. The upper bound shows that there are still a few errors by the assumption used by the 'monolingual' method. However, the high precision of the upper bound represents the well-formedness of the lexicon we used. We need more experiments on other bilingual lexicons to conclude that our method is available for

Features	Precision	Recall	F-score
All	0.911	0.956	0.932
$-h_{1F}, h_{1E}$	0.911	**0.974**	**0.941**
$-h_{2F}, h_{2E}$	0.906	0.947	0.926
$-h_{3F}, h_{3E}$	0.939	0.930	0.934
$-h_{4F}, h_{4E}$	0.919	0.734	0.816
$-h_{5F}, h_{5E}$	0.869	0.804	0.831
$-h_6, h_7$	**0.940**	0.934	0.937
$-$combs.	0.936	0.929	0.932

Table 7: Evaluation scores of the bilingual method with removing features on the development set
$-h$ represents removing the feature h and combinatorial features using h. $-$combs. represents removing all the combinatorial features.

many kinds of lexicons.

To investigate the effectiveness of each feature, we compared the scores when we remove several features. Table 7 shows these results. Contrary to our intuition, we found that features of agreement of the first characters (h_1) remarkably degraded the recall without gains in precision. One of the reasons for such results is that there are many cases of non-synonyms that have the same first character. We need to investigate more effective combinations of features or to apply other machine learning techniques for improving the performance. From these results, we consider that the features of h_4 are effective for improving the recall, and that the features of h_2 and h_5 contribute improvement of both the precision and the recall. h_3, h_6, h_7, and combinatorial features seem to improve the recall at the expense of precision. Which measure is important depends on the importance of our target for using this technique. It depends on the requirements that we emphasize, but in general the recall is more important for finding more bilingual synonyms.

5 Conclusion and future work

This paper proposed a method for identifying bilingual synonyms in a bilingual lexicon by using clues of spelling variations. We described the notion of bilingual synonyms, and presented two approaches for identifying them: one is to directly predict the relation, and another is to merge monolingual synonyms identified, according to the bilingual lexicon. Our experiments showed that the proposed method significantly outperformed the method that did not use features primarily with spelling variations; the proposed method extracted bilingual synonyms with high precision and recall. In addition, we found that merging monolingual synonyms by the dictionary is effective for finding bilingual synonyms; there occur few errors through the assumption described in Section 3.2.1.

Our future work contains implementing more features for identifying synonymous relations, constructing bilingual synonym sets, and evaluating our method for specific tasks such as thesaurus construction or cross-lingual information retrieval.

Currently, the features used do not include other clues with spelling variations, such as the weighted edit distance, transformation patterns, stemming and so on. Another important clue is distributional information, such as the context. We can use both monolingual and bilingual corpora for extracting distributions of terms, and bilingual corpora are expected to be especially effective for our goal.

We did not perform an experiment to construct bilingual synonym sets from synonym pairs in this paper. Described in Section 3.1, bilingual synonym sets can be constructed from bilingual synonym pairs by assuming some approximations. The approximation that permits transitivity of bilingual synonymous relations increases identified bilingual synonyms, and thus causes an increase in recall and decrease in precision. It is an open problem to find appropriate strategies for constructing bilingual synonym sets.

Finally, we plan to evaluate our method for specific tasks. For data-driven machine translation, it is expected that data sparseness problem is alleviated by merging the occurrences of low-frequency terms. Another application is cross-lingual information retrieval, which can be improved by using candidate expanded queries from bilingual synonym sets.

Acknowledgments

This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan) and Japanese/Chinese Machine Translation Project in Special Coordination Funds for Promoting Science and Technology (MEXT, Japan). We thank

Japan Science and Technology Agency (JST) for providing a useful bilingual lexicon with synonymous information. We acknowledge the anonymous reviewers for helpful comments and suggestions.

References

Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Vincent D. Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. 2004. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Review*, 46(4):647–666.

Carolyn J. Croach and Bokyung Yang. 1992. Experiments in automatic statistical thesaurus construction. In *Proc. of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–88. ACM Press.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Béatrice Daille, Benoît Habert, Christian Jacquemin, and Jean Royauté. 1996. Empirical observation of term variations and principles for their description. *Terminology*, 3(2):197–258.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *Proc. of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyser JUMAN. In *Proc. of International Workshop on Sharable Natural Language Resources*, pages 22–28.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms among distributionally similar words. In *Proc. of the 2003 International Joint Conference on Artificial Intelligence*, pages 1492–1493.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of the 17th International Conference on Computational Linguistics*, volume 2, pages 768–774.

Julie B. Lovins. 1968. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31.

Takeshi Masuyama, Satoshi Sekine, and Hiroshi Nakagawa. 2004. Automatic construction of Japanese KATAKANA variant list from large corpus. In *Proc. of the 20th International Conference on Computational Linguistics*, volume 2, pages 1214–1219.

Philippe Muller, Nabil Hathout, and Bruno Gaume. 2006. Synonym extraction using a semantic distance on a dictionary. In *Proc. of TextGraphs: the 2nd Workshop on Graph Based Methods for Natural Language Processing*, pages 65–72.

Young C. Park and Key-Sun Choi. 1996. Automatic thesaurus construction using Bayesian networks. *Information Processing and Management*, 32(5):543–553.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Yonggang Qiu and Hans-Peter Frei. 1993. Concept-based query expansion. In *Proc. of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 160–169.

Ariel S. Schwartz and Marti A. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Proc. of the 8th Pacific Symposium on Biocomputing*, pages 451–462.

Mitsuo Shimohata and Eiichiro Sumita. 2002. Automatic paraphrasing based on parallel corpus for normalization. In *Proc. of the 3rd International Conference on Language Resources and Evaluation*, volume 2, pages 453–457.

Scott A. Waterman. 1996. Distinguished usage. In *Corpus Processing for Lexical Acquisition*, pages 143–172. MIT Press.

Hua Wu and Ming Zhou. 2003. Optimizing synonym extraction using monolingual and bilingual resources. In *Proc. of the 2nd International Workshop on Paraphrasing*.

Minimally Supervised Multilingual Taxonomy and Translation Lexicon Induction

Nikesh Garera and **David Yarowsky**
Department of Computer Science
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218, USA
{ngarera,yarowsky}@cs.jhu.edu

Abstract

We present a novel algorithm for the acquisition of multilingual lexical taxonomies (including hyponymy/hypernymy, meronymy and taxonomic cousinhood), from monolingual corpora with minimal supervision in the form of seed exemplars using discriminative learning across the major WordNet semantic relationships. This capability is also extended robustly and effectively to a second language (Hindi) via cross-language projection of the various seed exemplars. We also present a novel model of translation dictionary induction via multilingual transitive models of hypernymy and hyponymy, using these induced taxonomies. Candidate lexical translation probabilities are based on the probability that their induced hyponyms and/or hypernyms are translations of one another. We evaluate all of the above models on English and Hindi.

1 Introduction

Taxonomy resources such as WordNet are limited or non-existent for most of the world's languages. Building a WordNet manually from scratch requires a huge amount of human effort and for rare languages the required human and linguistic resources may simply not be available. Most of the automatic approaches for extracting semantic relations (such as hyponyms) have been demonstrated for English and some of them rely on various language-specific resources (such as supervised training data, language-specific lexicosyntactic patterns, shallow parsers, etc.). This paper presents a language independent approach for inducing taxonomies such as shown in Figure 1 using limited supervision and linguistic resources. We propose a seed learning based approach for extracting semantic relations (hyponyms, meronyms and cousins) that improves upon existing induction frameworks by combining evidence from multiple semantic relation types. We show that using a joint model for extracting different semantic relations helps to induce more relation-specific patterns and filter out the generic patterns[1]. The pat-

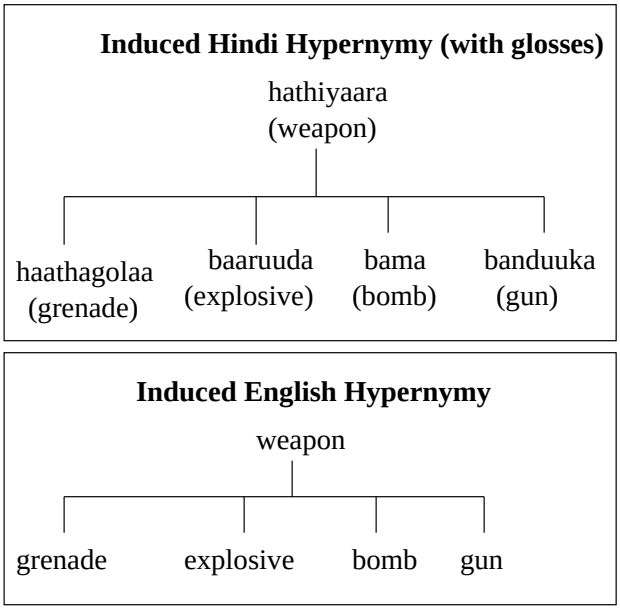


Figure 1: Goal: To induce multilingual taxonomy relationships in parallel in multiple languages (such as Hindi and English) for information extraction and machine translation purposes.

[1]By generic patterns, we mean patterns that cannot distinguish between different semantic relations. For example, the

terns can then be used for extracting new wordpairs expressing the relation. Note that the only training data used in the algorithm are the few seed pairs required to start the bootstrapping process, which are relatively easy to obtain. We evaluate the taxonomy induction algorithm on English and a second language (Hindi) and show that it can reliably and accurately induce taxonomies in two diverse languages.
We further show how having induced parallel taxonomies in two languages can be used for augmenting a translation dictionary between those two languages. We make use of the automatically induced hyponym/hypernym relations in each language to create a transitive "bridge" for dictionary induction. Specifically, the dictionary induction task relies on the key observation that words in two languages (e.g. English and Hindi) have increased probabilities of being translations of each other if their hypernyms or hyponyms are translations of one another.

2 Related Work

While manually created WordNets for English (Fellbaum, 1998) and Hindi (Narayan, 2002) have been made available, a lot of time and effort is required in building such semantic taxonomies from scratch. Hence several automatic corpus based approaches for acquiring lexical knowledge have been proposed in the literature. Much of this work has been done for English based on using a few evocative fixed patterns including "X and other Ys", "Y such as X", as in the classic work by Hearst (1992). The problems with using a few fixed patterns is the often low coverage of such patterns; thus there is a need for discovering additional informative patterns automatically. There has been a plethora of work in the area of information extraction using automatically derived patterns contextual patterns for semantic categories (e.g. companies, locations, time, person-names, etc.) based on bootstrapping from a small set of seed words (Riloff and Jones, 1999; Agichtein and Gravano, 2000; Thelen and Riloff, 2002; Ravichandran and Hovy, 2002; Hasegawa et al. 2004; Etzioni et al. 2005; Paşca et al. 2006). This framework has been also shown to work for extracting semantic relations between entities: Pantel et al. (2004) proposed an approach based on edit-distance to learn lexico-POS patterns for *is-a* and *part-of* relations. Girju et al. (2003) used 100 seed words from WordNet to extract patterns for *part-of* relations. While most of the above pattern induction work has been shown to work well for specific relations (such as "birthdates, companies, etc."), Section 3.1 explains why directly applying seed learning for semantic relations can result in high recall but low precision patterns, a problem also noted by Pantel and Pennacchiotti (2006). Furthermore, much of the semantic relation extraction work has focused on extracting a particular relation *independently* of other relations. We show how this problem can be solved by combining evidence from multiple relations in Section 3.2. Snow et al.(2006) also describe a probablistic framework for combining evidence using constraints from hyponymy and cousin relations. However, they use a supervised logistic regression model. Moreover, their features rely on parsing dependency trees which may not be available for most languages.
The key contribution of this work is using evidence from multiple relationship types in the seed learning framework for inducing these relationships and conducting a multilingual evaluation for the same. We further show how extraction of semantic relations in multiple languages can be applied to the task of improving a dictionary between those languages.

3 Approach

To be able to automatically create taxonomies such as WordNet, it is useful to be able to learn not only hyponymy/hyponymy directly, but also the additional semantic relationships of meronymy and taxonomic cousinhood. Specifically, given a pair of words (X, Y), the task is to answer the following questions: 1. Is X a hyponym of Y (e.g. *weapon, gun*)? 2. Is X a part/member of Y (e.g. *trigger, gun*)? 3. Is X a cousin/sibling[2] of Y (e.g. *gun, missile*)? 4. Do none of the above 3 relations apply but X is observed in the context of Y (e.g. *airplane,accident*)?[3] We will refer to class 4 as "other".

pattern "X and Y" is a generic pattern whereas the pattern "Y such as X" is a hyponym-specific pattern

[2]Cousins/siblings are words that share a close common hypernym

[3]Note that this does not imply X is unrelated or independent of Y. On the contrary, the required sentential co-occurence implies a topic similarity. Thus, this is a much harder class to distinguish from classes 1-3 than non co-occuring unrelatedness (such as *gun, protazoa*) and hence was included in the evaluation.

Rank	English	Hindi
1	Y, the X	Y aura X (Gloss: Y and X)
2	Y and X	Y va X (Gloss: Y in addition to X)
3	X and other Y	Y ne X (Gloss: Y (case marker) X)
4	X and Y	X ke Y (Gloss: X's Y)
5	Y, X	Y me.n X (Gloss: Y in X)

Table 1: Naive pattern scoring: Hyponymy patterns ranked by their raw corpus frequency scores.

Rank	English	Hindi
1	Y like X	X aura anya Y (Gloss: X and other Y)
2	Y such as X	Y, X (Gloss: Y, X)
3	X and other Y	X jaise Y (Gloss: X like Y)
4	Y and X	Y tathaa X (Gloss: Y or X)
5	Y, including X	X va anya Y (Gloss: X and other Y)

Table 2: Patterns for hypernymy class reranked using evidence from other classes. Patterns distributed fairly evenly across multiple relationship types (e.g. "X and Y") are deprecated more than patterns focused predominantly on a single relationship type (e.g. "Y such as X").

3.1 Independently Bootstrapping Lexical Relationship Models

Following the pattern induction framework of Ravichandran and Hovy (2002), one of the ways of extracting different semantic relations is to learn patterns for each relation independently using seeds of that relation and extract new pairs using the learned patterns. For example, to build an independent model of hyponymy using this framework, we collected approximately 50 seed exemplars of hyponym pairs and extracted all the patterns that match with the seed pairs[4]. As in Ravichandran and Hovy (2002), the patterns were ranked by corpus frequency and a frequency threshold was set to select the final patterns. These patterns were then used to extract new word pairs expressing the hyponymy relation by finding word pairs that occur with these patterns in an unlabeled corpus. However, the problem with this approach is that generic patterns (like "X and Y") occur many times in a corpus and thus low-precision patterns may end up with high cumulative scores. This problem is illustrated more clearly in Table 1, which shows a list of top five hyponymy patterns (ranked by their corpus frequency) using this approach. We overcome this problem by exploiting the multi-class nature of our task and combine evidence from multiple relations in order to learn high precision patterns (with high conditional probabilities) for each relation. The key idea is to weed out the patterns that occur in more than one semantic relation and keep the ones that are relation-specific[5], thus using the relations meronymy, cousins and other as *negative evidence* for hyponymy and vice versa. Table 2 shows the pattern ranking by using the model developed in Section 3.2 that makes use of evidence from different classes. We can see more hyponymy specific patterns ranked at the top[6] suggesting the usefulness of this method in finding class-specific patterns.

3.2 A minimally supervised multi-class classifier for identifying different semantic relations

First, we extract a list of patterns from an unlabeled corpus[7] independently for each relationship type (class) using the seeds[8] for the respective class as in Section 3.1.[9] In order to develop a multi-

[4] A pattern is the ngrams occurring between the seedpair (also called gluetext). The length of the pattern was thresholded to 15 words.

[5] In the actual algorithm, we will not be entirely weeding out the common patterns but will estimate the conditional class probabilities for each pattern: $p(class|pattern)$

[6] It is interesting to see in Table 2 that the top learned Hindi hyponymy patterns seem to be translations of the English patterns suggested by Hearst (1992). This leads to an interesting future work question: Are the most effective hyponym patterns in other languages usually translations of the English hyponym patterns proposed by Hearst (1992) and what are frequent exceptions?

[7] Unlabeled monolingual corpora were used for this task, the English corpus was the LDC Gigaword corpus and the Hindi corpus was newswire text extracted from the web containing a total of 64 million words.

[8] The number of seeds used for classes {hyponym, meronym, cousin, other} were {48,40,49,50} for English and were {32,58,31,35} for Hindi respectively. A sample of seeds used is shown in Table 5.

[9] We retained only the patterns that had seed frequency greater one for extracting new word pairs. The total number

	Hypo.	Mero.	Cous.	Other
X *of the Y*	0	0.66	0.04	0.3
Y, *especially X*	1	0	0	0
Y, *whose X*	0	1	0	0
X *and other Y*	0.63	0.08	0.18	0.11
X *and Y*	0.23	0.3	0.33	0.14

Table 3: A sample of patterns and their relationship type probabilities $P(class|pattern)$ extracted at the end of training phase for English.

	Hypo.	Mero.	Cous.	Other
X *aura anya Y* (X and other Y)	1	0	0	0
X *aura Y* (X and Y)	0.09	0.09	0.71	0.11
X *jaise Y* (X like Y)	1	0	0	0
X *va Y* (X and Y)	0.11	0	0.89	0
Y *kii X* (Y's X)	0.33	0.67	0	0

Table 4: A sample of patterns and their class probabilities $P(class|pattern)$ extracted at the end of training phase for Hindi.

class probabilistic model, we obtain the probability of each class c given the pattern p as follows:

$$P(c|p) = \frac{seed_{freq}(p,c)}{\sum_{c'} seed_{freq}(p,c')}$$

where $seed_{freq}(p, c)$ is the number of seeds of class c that were found with the pattern p in an unlabeled corpus. A sample of the $P(class|pattern)$ tables for English and Hindi are shown in the Tables 3 and 4 respectively. It is clear how occurrence of a pattern in multiple classes can be used for finding reliable patterns for a particular class. For example, in Table 3: although the pattern "X and Y" will get a higher seed frequency than the pattern "Y, especially X", the probability $P(\text{“}X\ and\ Y''|hyponymy)$ is much lower than $P(\text{“}Y,\ especially\ X''|hyponymy)$, since the pattern "Y, especially X" is unlikely to occur with seeds of other relations.

Now, instead of using the $seed_{freq}(p, c)$ as the score for a particular pattern with respect to a class, we can rescore patterns using the probabilities $P(class|pattern)$. Thus the final score for a pattern p with respect to class c is obtained as:

$$score(p, c) = seed_{freq}(p, c) \cdot P(c|p)$$

We can view this equation as balancing recall and precision, where the first term is the frequency of the pattern with respect to seeds of class c (representing recall), and the second term represents the relation-specificness of the pattern with respect to class c (representing precision). We recomputed the score for each pattern in the above manner and obtain a ranked list of patterns for each of the classes for English and Hindi. Now, to extract new pairs for each class, we take all the patterns with a seed frequency greater than 2 and use them to extract word pairs from an unlabeled corpus. The semantic class for each extracted pair is then predicted using the multi-class classifier as follows: Given a pair of words (X1, X2), note all the patterns that matched with this pair in the unlabeled corpus, denote this set as $\mathcal{P}$. Choose the predicted class $c*$ for this pair as:

$$c^* = argmax_c \sum_{p \in \mathcal{P}} score(p, c)$$

3.3 Evaluation of the Classification Task

Over 10,000 new word relationship pairs were extracted based on the above algorithm. While it is hard to evaluate all the extracted pairs manually, one can certainly create a representative smaller test set and evaluate performance on that set. The test set was created by randomly identifying word pairs in WordNet and newswire corpora and annotating their correct semantic class relationships. Test set construction was done entirely independently from the algorithm application, and hence some of the test pairs were missed entirely by the learning algorithm, yielding only partial coverage.

The total number of test examples including all classes were 200 and 140 for English and Hindi test-sets respectively. The overall coverage[10] on these test-sets was 81% and 79% for English and Hindi respectively. Table 6 reports the overall accuracy[11] for the 4-way classification using different patterns scoring methods. Baseline 1 is scoring patterns by their corpus frequency as in Ravichandran and Hovy (2002), Baseline 2 is another intutive method of

of retained patterns across all classes for {English,Hindi} were {455,117} respectively.

[10] Coverage is defined as the percentage of the test cases that were present in the unlabeled corpus, that is, cases for which an answer was given.

[11] Accuracy on a particular set of pairs is defined as the percentage of pairs in that set whose class was correctly predicted.

	English		Hindi	
	Seed Pairs	Model Predictions	Seed Pairs	Model Predictions
Hypernym	tool,hammer	gun,weapon	khela,Tenisa (game,tennis)	kaa.ngresa,paarTii (congress,party)
	currency,yen	hockey,sport	appraadha,hatyaa (crime,murder)	passporTa,kaagajaata (passport,document)
	metal,copper	cancer,disease	jaanvara,bhaaga (animal,tiger)	a.ngrejii,bhaashhaa (English,language)
Meronym	wheel,truck	room,hotel	u.ngalii,haatha (finger,hand)	jeba,sharTa (pocket,shirt)
	headline,newspaper	bark,tree	kamaraa,aspataala (room,hospital)	kaptaana,Tiima (captain,team)
	wing,bird	lens,camera	ma.njila,imaarata (floor,building)	darvaaja,makaana (door,house)
Cousin	dollar,euro	guitar,drum	bhaajapa,kaa.ngresa (bjp,congress)	peTrola,Diijala (petrol,diesel)
	heroin,cocaine	history, geography	Hindii,a.ngrejii (Hindi,English)	Daalara,rupayaa (dollar,rupee)
	helicopter,submarine	diabetes,arthritis	basa,Traka (bus,truck)	talaaba,nadii (pond,river)

Table 5: A sample of seeds used and model predictions for each class for the taxonomy induction task. For each of the model predictions shown above, its Hyponym/Meronym/Cousin classification was correctly assigned by the model.

scoring patterns by the number of seeds they extract. The third row in Table 6 indicates the result of rescoring patterns by their class conditional probabilties, giving the best accuracy.

While this method yields some improvement over other baselines, the main point to note here is that the pattern-based methods which have been shown to work well for English also perform reasonably well on Hindi, inspite of the fact that the size of the unlabeled corpus available for Hindi was 15 times smaller than for English.

Table 7 shows detailed accuracy results for each relationship type using the model developed in section 3.2. It is also interesting to see in Table 8 that most of the confusion is due to "other" class being classified as "cousin" which is expected as cousin words are only weakly semantically related and uses more generic patterns such as "X and Y" which can often be associated with the "other" class as well. Strongly semantically clear classes like Hypernymy and Meronymy seem to be well discriminated as their induced patterns are less likely to occur in other relationship types.

Model	English Accuracy	Hindi Accuracy
Baseline 1 $_{[RH02]}$	65%	63%
Baseline 2 $_{seed_{freq}}$	70%	65%
$seed_{freq} \cdot P(c\|p)$	**73%**	**66%**

Table 6: Overall accuracy for 4-way classification {hypernym,meronym,cousin,other} using different pattern scoring methods.

	English			Hindi		
	Total	Cover.	Acc.	Total	Cover.	Acc.
Hypr.	83	74%	97%	59	82%	75%
Mero.	41	81%	88%	33	63%	81%
Cous.	42	91%	55%	23	91%	71%
Other	34	85%	31%	25	80%	20%
Overall	200	81%	73%	140	79%	66%

Table 7: Test set coverage and accuracy results for inducing different semantic relationship types.

	English				Hindi			
	Hypo.	Mero.	Cous.	Oth.	Hypo.	Mero.	Cous.	Oth.
Hypo.	59	1	1	0	36	1	10	1
Mero.	1	28	1	3	0	17	4	0
Cous.	14	3	21	0	6	0	15	0
Other	7	3	10	9	1	4	11	4

Table 8: Confusion matrix for English (left) Hindi (right) for the four-way classification task

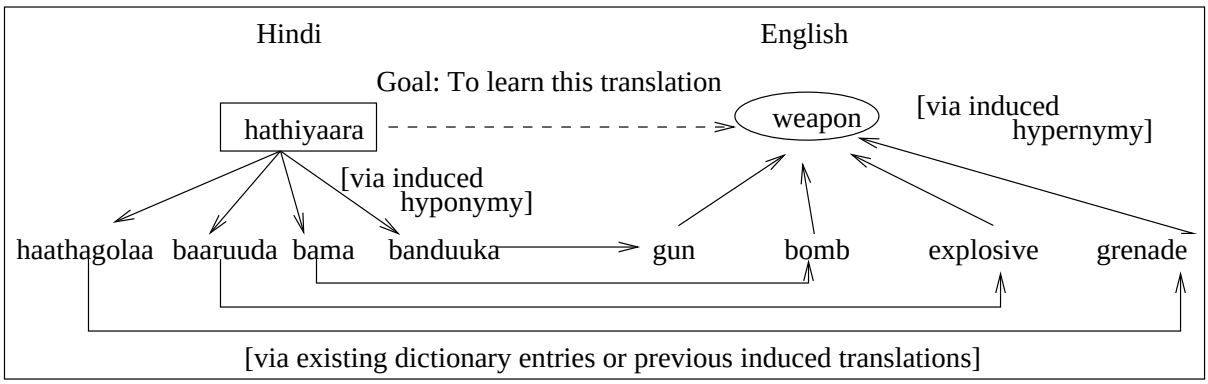


Figure 2: Illustration of the models of using induced hyponymy and hypernymy for translation lexicon induction.

4 Improving a partial translation dictionary

In this section, we explore the application of automatically generated multilingual taxonomies to the task of translation dictionary induction. The hypothesis is that a pair of words in two languages would have increased probability of being translations of each other if their hypernyms or hyponyms are translations of one another.

As illustrated in Figure 2, the probability that *weapon* is a translation of the Hindi word *hathiyaara* can be decomposed into the sum of the probabilities that their hyponyms in both languages (as induced in Section 3.2) are translations of each other. Thus:

$$P_{H->E}\ (W_E|W_H) = \sum_i P_{hyper}\ (W_E|Eng(H_i))\ P_{hypo}(H_i|W_H)$$

for induced hyponyms H_i of the source word W_H, and using an existing (and likely very incomplete) Hindi-English dictionary to generate Eng(H_i) for these hyponyms, and the corresponding induced hypernyms of these translations in English.[12]. We conducted a very preliminary evaluation of this idea for obtaining English translations of a set of 25 Hindi words. The Hindi candidate hyponym space had been pruned of function words and non-noun words. The likely English translation candidates for each Hindi word were ranked according to the probability $P_{H->E}(W_E|W_H)$.

The first column of Table 9 shows the stand-alone performance for this model on the dictionary induction task. This standalone model has a reasonably good accuracy for finding the correct translation in the Top 10 and Top 20 English candidates.

	Accuracy (uni-d)	Accuracy (bi-d)	Accuracy bi-d + Other
Top 1	20%	36%	36%
Top 5	56%	64%	72%
Top 10	72%	72%	80%
Top 20	84%	84%	84%

Table 9: Accuracy on Hindi to English word translation using different transitive hypernym algorithms. The additional model components in the bi-d(irectional) plus Other model are only used to rerank the top 20 candidates of the bidirectional model, and are hence limited to its top-20 performance.

This approach can be further improved by also implementing the above model in the reverse direction and computing the $P(W_H|W_{E_i})$ for each of the English candidates E_i. We did so and computed $P(W_H|W_{E_i})$ for top 20 English candidate translations. The final score for an English candidate translation given a Hindi word was combined by a simple average of the two directions, that is, by summing $P(W_{E_i}|W_H)$ + $P(W_H|W_{E_i})$.

The second column of Table 9 shows how this bidirectional approach helps in getting the right

[12]One of the challenges of inducing a dictionary via using a corpus based taxonomy is sense disambiguation of the words to be translated. In the current model, the more dominant sense (in terms of corpus frequency of its hyponyms) is likely to get selected by this approach. While the current model can still help in getting translations of the dominant sense, possible future work would be to cluster all the hyponyms according to contextual features such that each cluster can represent the hyponyms for a particular sense. The current dictionary induction model can then be applied again using the hyponym clusters to distinguish different senses for translation.

translations in Top 1 and Top 5 as compared to the unidirectional approach. Table 10 shows a sample

Correctly translated	Incorrectly translated
aujaara (tool)	vishaya (topic)
biimaarii (disease)	saamana (stuff)
hathiyaara (weapon)	dala (group,union)
dastaaveja (documents)	tyohaara (festival)
aparaadha (crime)	jagaha (position,location)

Table 10: A sample of correct and incorrect translations using transitive hypernymy/hyponym word translation induction

of correct and incorrect translations generated by the above model. It is interesting to see that the incorrect translations seem to be the words that are very general (like "topic", "stuff", etc.) and hence their hyponym space is very large and diffuse, resulting in incorrect translations.While the columns 1 and 2 of Table 9 show the standalone application of our translation dictionary induction method, we can also combine our model with existing work on dictionary induction using other translation induction measures such as using relative frequency similarity in multilingual corpora and using cross-language context similarity between word co-occurrence vectors (Schafer and Yarowsky, 2002).We implemented the above dictionary induction measures and combined the taxonomy based dictionary induction model with other measures by just summing the two scores[13]. The preliminary results for bidirectional hypernym/hyponym + other features are shown in column 3 of Table 9. The results show that the hypernym/hyponym features can be a useful orthogonal source of lexical similarity in the translation-induction model space. While the model shown in Figure 2 proposes inducing translations of hypernyms, one can also go in the other direction and induce likely translation candidates for hyponyms by knowing the translation of hypernyms. For example, to learn that *rifle* is a likely translation candidate of the Hindi word *raaiphala*, is illustrated in Figure 3. But because there is a much larger space of hyponyms for *weapon* in this direction, the output serves more to reduce the entropy of the translation candidate space when used in conjunction with other translation induction similarity measures. We would expect the application of additional similarity measures to this greatly narrowed and ranked hypothesis space to yield improvement in future work.

[13]after renormalizing each of the individual score to be in the range 0 to 1.

5 Conclusion

This paper has presented a novel minimal-resource algorithm for the acquisition of multilingual lexical taxonomies (including hyponymy/hypernymy and meronymy). The algorithm is based on cross language projection of various monolingual indicators of these taxonomic relationships in free text and via bootstrapping thereof. Using only 31-58 seed examples, the algorithm achieves accuracies of 73% and 66% for English and Hindi respectively on the tasks of hyponymy/meronomy/cousinhood/other model induction. The robustness of this approach is shown by the fact that the unannotated Hindi development corpus was only 1/15th the size of the utilized English corpus. We also present a novel model of unsupervised translation dictionary induction via multilingual transitive models of hypernymy and hyponymy, using these induced taxonomies and evaluated on Hindi-English. Performance starting from no multilingual dictionary supervision is quite promising.

References

E. Agichtein and L. Gravano. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 85–94.

M. J. Cafarella, D. Downey, S. Soderland, and O. Etzioni. 2005. Knowitnow: Fast, scalable information extraction from the web. In *Proceedings of EMNLP/HLT-05*, pages 563–570.

S. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of ACL-99*, pages 120–126.

B. Carterette, R. Jones, W. Greiner, and C. Barr. 2006. N semantic classes are harder than two. In *Proceedings of ACL/COLING-06*, pages 49–56.

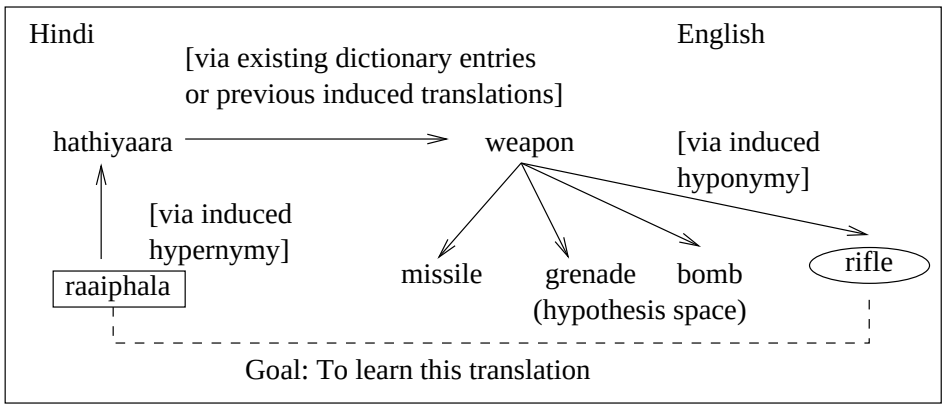


Figure 3: Reducing the space of likely translation candidates of the word *raaiphala* by inducing its hypernym, using a partial dictionary to look up the translation of hypernym and generating the candidate translations as induced hyponyms in English space.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165(1):91–134.

C. Fellbaum. 1998. *WordNet: An electronic lexical database.*

R. Girju, A. Badulescu, and D. Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In Proceedings of HLT/NAACL-03, *pages 1–8.*

R. Girju, A. Badulescu, and D. Moldovan. 2006. Automatic discovery of part-whole relations. Computational Linguistics, *21(1):83–135.*

D. Graff, J. Kong, K. Chen, and K. Maeda. 2005. English Gigaword Second Edition. Linguistic Data Consortium, catalog number LDC2005T12.

T. Hasegawa, S. Sekine, and R. Grishman. 2004. Discovering relations among named entities from large corpora. In Proceedings of ACL-04, *pages 415–422.*

M. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In Proceedings of COLING-92, *pages 539–545.*

D. Narayan, D. Chakrabarty, P. Pande, and P. Bhattacharyya. 2002. An Experience in Building the Indo WordNet-a WordNet for Hindi. International Conference on Global WordNet.

Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Names and similarities on the web: Fact extraction in the fast lane. In Proceedings of ACL/COLING-06, *pages 809–816.*

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In Proceedings of ACL/COLING-06, *pages 113–120.*

P. Pantel and D. Ravichandran. 2004. Automatically labeling semantic classes. In Proceedings of HLT/NAACL-04, *pages 321–328.*

P. Pantel, D. Ravichandran, and E. Hovy. 2004. Towards terascale knowledge acquisition. In Proceedings of COLING-04.

D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In Proceedings of ACL-02, *pages 41–47.*

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In Proceedings of AAAI/IAAI-99, *pages 474–479.*

E. Riloff and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. CoRR, *cmp-lg/9706013.*

C. Schafer and D. Yarowsky. 2002. Inducing translation lexicons via diverse similarity measures and bridge languages. In Proceedings of CONLL-02, *pages 146–152.*

R. Snow, D. Jurafsky, and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In Proceedings of ACL/COLING-06, *pages 801–808.*

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In Proceedings of EMNLP-02, *pages 214–221.*

D. Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In Proceedings of HLT/NAACL-03, *pages 197–204.*

Japanese-Spanish Thesaurus Construction

Using English as a Pivot

Jessica Ramírez, Masayuki Asahara, Yuji Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
Ikoma, Nara, 630-0192 Japan
{jessic-r,masayu-a,matsu}@is.naist.jp

Abstract

We present the results of research with the goal of automatically creating a multilingual thesaurus based on the freely available resources of Wikipedia and WordNet. Our goal is to increase resources for natural language processing tasks such as machine translation targeting the Japanese-Spanish language pair. Given the scarcity of resources, we use existing English resources as a pivot for creating a trilingual Japanese-Spanish-English thesaurus. Our approach consists of extracting the translation tuples from Wikipedia, disambiguating them by mapping them to WordNet word senses. We present results comparing two methods of disambiguation, the first using VSM on Wikipedia article texts and WordNet definitions, and the second using categorical information extracted from Wikipedia, We find that mixing the two methods produces favorable results. Using the proposed method, we have constructed a multilingual Spanish-Japanese-English thesaurus consisting of 25,375 entries. The same method can be applied to any pair of languages that are linked to English in Wikipedia.

1 Introduction

Aligned data resources are indispensable components of many Natural Language Processing (NLP) applications; however lack of annotated data is the main obstacle for achieving high performance NLP systems. Current success has been moderate. This is because for some languages there are few resources that are usable for NLP.

Manual construction of resources is expensive and time consuming. For this reason NLP researchers have proposed semi-automatic or automatic methods for constructing resources such as dictionaries, thesauri, and ontologies, in order to facilitate NLP tasks such as word sense disambiguation, machine translation and other tasks. Hoglan Jin and Kam-Fai Wong (2002) automatically construct a Chinese dictionary from different Chinese corpora, and Ahmad Khurshid et al. (2004) automatically develop a thesaurus for a specific domain by using text that is related to an image collection to aid in image retrieval.

With the proliferation of the Internet and the immense amount of data available on it, a number of researchers have proposed using the World Wide Web as a large-scale corpus (Rigau et al., 2002). However due to the amount of redundant and ambiguous information on the web, we must find methods of extracting only the information that is useful for a given task.

1.1 Goals

This research deals with the problem of developing a multilingual Japanese-English-Spanish thesaurus that will be useful to future Japanese-Spanish NLP research projects.

A thesaurus generally means a list of words grouped by concepts; the resource that we create is similar because we group the words according to semantic relations. However, our resource is also

composed of three languages – Spanish, English, and Japanese. Thus we call the resource we created a multilingual thesaurus.

Our long term goal is the construction of a Japanese-Spanish MT system. This thesaurus will be used for word alignments and building comparable corpus.

We construct our multilingual thesaurus by following these steps:

- Extract the translation tuples from Wikipedia article titles
- Align the word senses of these tuples with those of English WordNet (disambiguation)
- Construct a parallel thesaurus of Spanish-English-Japanese from these tuples

1.2 Method summary

We extract the translation tuples using Wikipedia's hyperlinks to articles in different languages and align these tuples to WordNet by measuring cosine vector similarity measures between Wikipedia article texts and WordNet glosses. We also use heuristics comparing the Wikipedia categories of a word with its hypernyms in WordNet.

A fundamental step in the construction of a thesaurus is part of speech (POS) identification of words and word sense disambiguation (WSD) of polysemous entries.

For POS identification, we cannot use Wikipedia, because it does not contain POS information. So we use another well-structured resource, WordNet, to provide us with the correct POS for a word.

These two resources, Wikipedia and WordNet, contain polysemous entries. We also introduce WSD method to align these entries.

We focus on the multilingual application of Wikipedia to help transfer information across languages. This paper is restricted mainly to nouns, noun phrases, and to a lesser degree, named entities, because we only use Wikipedia article titles.

2 Resources

2.1 Wikipedia

Wikipedia is an online multilingual encyclopedia with articles on a wide range of topics, in which the texts are aligned across different languages.

Wikipedia has some features that make it suitable for research such as:

Each article has a title, with a unique ID. "Redirect pages" handle synonyms, and "disambiguation pages" are used when a word has several senses. "Category pages" contain a list of words that share the same semantic category. For example the category page for "Birds" contains links to articles like "parrot", "penguin", etc. Categories are assigned manually by users and therefore not all pages have a category label.

Some articles belong to multiple categories. For example, the article "Dominican Republic" belongs to three categories: "Dominican Republic", "Island countries" and "Spanish-speaking countries". Thus, the article Dominican Republic appears in three different category pages.

The information in redirect pages, disambiguation pages and Category pages combines to form a kind of Wikipedia taxonomy, where entries are identified by semantic category and word sense.

2.2 WordNet

WordNet (C. Fellbaum, 1998) "*is considered to be one of the most important resources in computational linguistics and is a lexical database, in which concepts have been grouped into sets of synonyms (words with different spellings, but the same meaning), called synsets, recording different semantic relations between words".*

WordNet can be considered to be a kind of machine-readable dictionary. The main difference between WordNet and conventional dictionaries is that WordNet groups the concepts into *synsets*, and each concept has a small definition sentence call a "gloss" with one or more sample sentences for each *synset*.

When we look for a word in WordNet it presents a finite number of synsets, each one representing a concept or idea.

The entries in WordNet have been classified according to the syntactic category such as: nouns, verbs, adjectives and adverbs, etc. These syntactic categories are known as part of speech (POS).

3 Related Work

Compared to well-established resources such as WordNet, there are currently comparatively fewer researchers using Wikipedia as a data resource in

NLP. There are, however, works showing promising results.

The work most closely related to this paper is (M. Ruiz et al., 2005), which attempts to create an ontology by associating the English Wikipedia links with English WordNet. They use the "Simple English Wikipedia" and WordNet version 1.7 to measure similarity between concepts. They compared the WordNet glosses and Wikipedia by using the Vector Space Model, and presented results using the cosine similarity.

Our approach differs in that we disambiguate the Wikipedia category tree using WordNet hyper-/hyponym tree. We compare our approach to M. Ruiz et al., (2005) using it as the baseline in section 7.

Oi Yee Kwong (1998) integrates different resources to construct a thesaurus by using WordNet as a pivot to fill gaps between thesaurus and a dictionary.

Strube and Ponzetto (2006) present some experiments using Wikipedia for the computing semantic relatedness of words (a measure of degree to which two concepts are related in a taxonomy measured using all semantic relations), and compare the results with WordNet. They also integrate Google hits, in addition to Wikipedia and WordNet based measures.

4 General Description

First we extract from Wikipedia all the aligned links i.e. Wikipedia article titles. We map these on to WordNet to determine if a word has more than one sense (polysemous) and extract the ambiguous articles. We use two methods to disambiguate by assigning the WordNet sense to the polysemous words, we use two methods:

- Measure the cosine similarity between each Wikipedia article's content and the WordNet glosses.
- Compare the Wikipedia category to which the article belongs with the corresponding word in WordNet's ontology

Finally, we substitute the target word into Japanese and Spanish.

5 Extracting links from Wikipedia

The goal is the acquisition of Japanese-Spanish-English tuples of Wikipedia's article titles. Each Wikipedia article provides links to corresponding articles in different languages.

Every article page in Wikipedia has on the left hand side some boxes labeled: 'navigation', 'search', 'toolbox' and finally 'in other languages'. This has a list of all the languages available for that article, although the articles in each language do not all have exactly the same contents. In most cases English articles are longer or have more information than their counterparts in other languages, because the majority of Wikipedia collaborators are native English speakers.

Pre-processing procedure:

Before starting with the above phases, we first eliminate the irrelevant information from Wikipedia articles, to make processing easy and faster.The steps applied are as follows:

1. Extract the Wikipedia web articles
2. Remove from the pages all irrelevant information, such as images, menus, and special markup such as: "()", """, "*", etc...
3. Verify if a link is a redirected article and extract the original article
4. Remove all stopwords and function words that do not give information about a specific topic such as "the", "between", "on", etc.

Methodology

Figure 1. The article "bird" in English, Spanish and Japanese

Take all articles titles that are nouns or named entities and look in the articles' contents for the box

called '*In other languages*'. Verify that it has at least one link. If the box exists, it links to the same article in other languages. Extract the titles in these other languages and align them with the original article title.

For instance, Figure 1. shows the English article titled "bird" translated into Spanish as "ave", and into Japanese as "chourui" (鳥類). When we click Spanish or Japanese '*in other languages*' box, we obtain an article about the same topic in the other language. This gives us the translation as its title, and we proceed to extract it.

6 Aligning Wikipedia entries to WordNet senses

The goal of aligning English Wikipedia entries to WordNet 2.1 senses is to disambiguate the polysemous words in Wikipedia by means of comparison with each sense of a given word existing in WordNet.

A gloss in WordNet contains both an association of POS and word sense. For example, the entry "bark#n#1" is different than "bark#v#1" because their POSes are different. In this example, "n" denotes noun and "v" denotes verb. So when we align a Wikipedia article to a WordNet gloss, we obtain both POS and word sense information.

Methodology

We assign WordNet senses to Wikipedia's polysemous articles. Firstly, after extracting all links and their corresponding translations in Spanish and Japanese, we look up the English words in WordNet and count the number of senses that each word has. If the word has more than one sense, the word is polysemous.

We use two methods to disambiguate the ambiguous articles, the first uses cosine similarity and the second uses Wikipedia's category tree and WordNet's ontology tree.

6.1 Disambiguation using Vector Space Model

We use a Vector Space Model (VSM) on Wikipedia and WordNet to disambiguate the POS and-word sense of Wikipedia article titles. This gives us a correspondence to a WordNet gloss.

$$\cos \theta = \frac{V_1 \cdot V_2}{|V_1| . |V_2|}$$

Where V_1 represents the Wikipedia article's word vector and V_2 represents the WordNet gloss' word vector.

In order to transfer the POS and word sense information, we have to measure similarity metric between a Wikipedia article and a WordNet gloss.

Background

VSM is an algebraic model, in which we convert a Wikipedia article into a vector and compares it to a WordNet gloss (that has also been converted into a vector) using the cosine similarity measure. It takes the set of words in some Wikipedia article and compares them with the set of words of WordNet gloss. Wikipedia articles which have more words in common are considered similar documents.

In Figure 2 shows the vector of the word "bank", we want to compare the similitude between the Wikipedia article "bank-1" with the English WordNet "bank-1" and "bank-2".

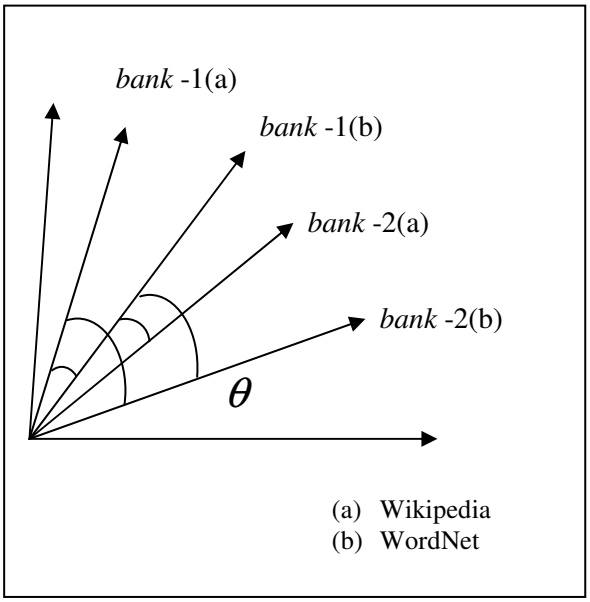


Figure 2. Vector Space Model with the word "bank"

VSM Algorithm:

1. Encode the Wikipedia article as a vector, where each dimension represents a word in the text of the article
2. Encode the WordNet gloss of each sense as a vector in the same manner

3. Compute the similarity between the Wikipedia vector and WordNet senses' vectors for a given word using the cosine measure
4. Link the Wikipedia article to the WordNet gloss with the highest similarity

6.2 Disambiguation by mapping the WordNet ontological tree to Wikipedia categories

This method consists of mapping the Wikipedia Category tree to the WordNet ontological tree, by comparing hypernyms and hyponyms. The main assumption is that there should be overlap between the hypernyms and hyponyms of Wikipedia articles and their correct WordNet senses. We will refer to this method as MCAT ("Map CATegories") throughout the rest of this paper.

Wikipedia has in the bottom of each page a box containing the category or categories to which the page belongs, as we can see in Figure 3. Each category links to the corresponding category page to which the title is affiliated. This means that the "category page" contains a list of all articles that share a common category.

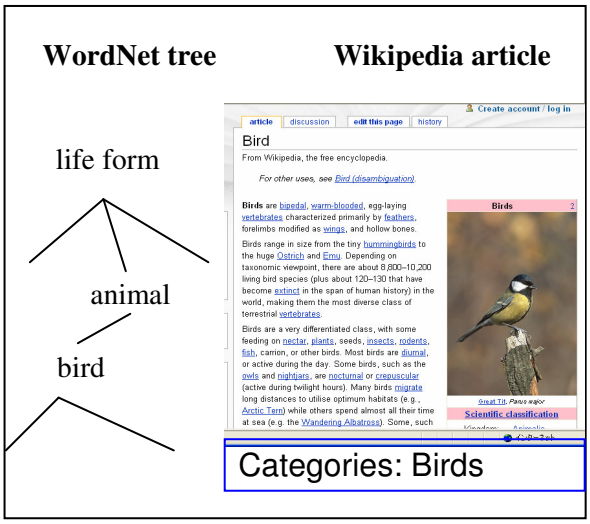


Figure 3. Relation between WordNet ontological tree and Wikipedia categories

Methodology

1. We extract ambiguous Wikipedia article titles (links) and the corresponding category pages
2. Extract the category pages, containing all pages which belong to that category, its subcategories, and other category pages that have a branch in the tree and categories to which it belongs.
3. If the page has a category:

3.1 Construct an n-dimensional vector containing the links and their categories

3.2 Construct an n-dimensional vector of the category pages, where every dimension represents a link which belongs to that category

4. For each category that an article belongs to:

4.1 Map the categoryto the WordNet hypernym-/hyponym tree by looking in each place that the given word appears and verify if any of its branches exist in the category page vector.

4.2 If a relation cannot be found then continue with other categories

4.3 If there is no correspondence at all then take the category pages vector and look to see if any of the links has relation with the WordNet tree

5. If there is at least one correspondence then assign this sense

6.3 Constructing the multilingual thesaurus

After we have obtained the English words with its corresponding English WordNet sense aligned in the three languages, we construct a thesaurus from these alignments.

The thesaurus contains a unique ID for every tuple of word and POS that it will have information about the syntactic category.

It also contains the sense of the word (obtain in the disambiguation process) and finally a small definition, which have the meaning of the word in the three languages.

- We assign a unique ID to every tuple of words
- For Spanish and Japanese we assign for default sense 1 to the first occurrence of the word if there exists more than 1 occurrence we continue incrementing
- Extract a small definition from the corresponding Wikipedia articles

The definition of title word in Wikipedia tends to be in the first sentence of the article.

Wikipedia articles often include sentences defining the meaning of the article's title. We mine Wikipedia for these sentences include them in our thesaurus. There is a large body of research dedicate to identifying definition sentences (Wilks et al., 1997), However, we currently rely on very simple patterns to this (e.g. "X is a/are Y", "X es un/a Y", "X は/が Y である。"). Incorporating more sophisticated methods remains an area of future work.

7 Experiments

7.1 Extracting links from Wikipedia

We use the articles titles from Wikipedia which are mostly nouns (including named entities) in Spanish, English and Japanese; (es.wikipedia.org, en.wikipedia.org, and ja.wikipedia.org), specifically "the latest all titles" and "the latest pages articles" files retrieved in April of 2006, and English WordNet version 2.1.

Our Wikipedia data contains a total of 377,621 articles in Japanese; 2,749,310 in English; and 194,708 in Spanish. We got a total of 25,379 words aligned in the three languages.

7.2 Aligning Wikipedia entries to WordNet senses

In WordNet there are 117,097 words and 141,274 senses. In Wikipedia (English) there are 2,749,310 article titles. 78,247 word types exist in WordNet. There are 14,614 polysemous word types that will align with one of the 141,274 senses in WordNet.

We conduct our experiments using 12,906 ambiguous articles from Wikipedia.

Table 1 shows the results obtained for WSD. The first column is the baseline (M. Ruiz et al., 2005) using the whole article; the second column is the baseline using only the first part of the article.

The third column (MCAT) shows the results of the second disambiguation method (disambiguation by mapping the WordNet ontological tree to Wikipedia categories). Finally the last column shows the results of combined method of taking the MCAT results when available and falling back to MCAT otherwise. The first row shows the sense assignments, the second row shows the incorrect sense assignment, and the last row shows the number of word used for testing.

7.2.1 Disambiguation using VSM

In the experiment using VSM, we used human evaluation over a sample of 507 words to verify if a given Wikipedia article corresponds to a given WordNet gloss. We took a the stratified sample of our data selecting the first 5 out of every 66 entries as ordered alphabetically for a total of 507 entries.

We evaluate the effectiveness of using whole articles in Wikipedia versus only a part (the first part up to the first subtitle), we found that the best score was obtained when using the whole articles 81.5% (410 words) of them are correctly assigned and 18.5% (97 words) incorrect.

Discussion

In this experiment because we used VSM the result was strongly affected by the length of the glosses in WordNet, especially in the case of related definitions because the longer the gloss the greater the probability of it having more words in common.

An example of related definitions in English WordNet is the word "apple". It has two senses as follows:

- apple#n#1: fruit with red or yellow or green skin and sweet to tart crisp whitish flesh.
- apple#n#2: native Eurasian tree widely cultivated in many varieties for its firm rounded edible fruits.

The Wikipedia article "apple" refers to both senses, and so selection of either WordNet sense is correct. It is very difficult for the algorithm to distinguish between them.

7.2.2 Disambiguation by mapping the WordNet ontological tree to Wikipedia categories

Our 12,906 articles taken from Wikipedia belong to a total of 18,810 associated categories. Thus, clearly some articles have more than one category; however some articles also do not have any category.

In WordNet there are 107,943 hypernym relations.

	Baseline		Our methods	
	VSM	VSM (using first part of the article)	MCAT	VSM+ MCAT
Correct sense identification	410 (81.5%)	403 (79.48%)	380 (95%)	**426 (84.02%)**
Incorrect sense identification	97 (18.5%)	104 (20.52%)	20 (5%)	81 (15.98%)
Total ambiguous words	507 (100%)		400 (100%)	507 (100%)

Table 1. Results of disambiguation

Results:

We successfully aligned 2,239 Wikipedia article titles with a WordNet sense. 400 of the 507 articles in our test data have Wikipedia category pages allowing us apply MCAT. Our human evaluation found that 95% (380 words) were correctly disambiguated. This outperformed disambiguation using VSM, demonstrating the utility of the taxonomic information in Wikipedia and WordNet. However, because not all words in Wikipedia have categories, and there are very few named entities in WordNet, the number of disambiguated words that can be obtained with MCAT (2,239) is less than when using VSM, (12,906).

Using only MCAT reduces the size of the Japanese-Spanish thesaurus. We had the intuition that by combining both disambiguation methods we can achieve a better balance between coverage and accuracy. VSM+MCAT use the MCAT WSD results when available falling back to VSM results otherwise.

We got an accuracy of 84.02% (426 of 507 total words) with VSM+MCAT, outperforming the baselines.

Evaluating the coverage over Comparable corpus

- Corpus construction

We construct comparable corpus by extracting from Wikipedia articles content information as follows:
Choose the articles whose content belongs to the thesaurus. We only took the first part of the article until a subtitle and split into sentences.

- Evaluation of coverage

We evaluate the coverage of the thesaurus over an automated comparable corpus automatically extracted from Wikipedia. The comparable corpus consists of a total of 6,165 sentences collected from 12,900 articles of Wikipedia.

We obtained 34,525 types of words; we map them with 15,764 from the Japanese-English-Spanish thesaurus. We found 10,798 types of words that have a coincidence that it is equivalent to 31.27%.

We found this result acceptable for find information inside Wikipedia.

8 Conclusion and future work

This paper focused on the creation of a Japanese-Spanish-English thesaurus and ontological relations. We demonstrated the feasibility of using Wikipedia's features for aligning several languages.We present the results of three sub-tasks:

The first sub-task used pattern matching to align the links between Spanish, Japanese, and English articles' titles.

The second sub-task used two methods to disambiguate the English article titles by assigning the WordNet senses to each English word; the first method compares the disambiguation using cosine similarity. The second method uses Wikipedia categories. We established that using Wikipedia categories and the WordNet ontology gives promising results, however the number of words that can be disambiguated with this method

is small compared to the VSM method. However, we showed that combining the two methods achieved a favorable balance of coverage and accuracy.

Finally, the third sub-task involved translating English thesaurus entries into Spanish and Japanese to construct a multilingual aligned thesaurus.

So far most of research on Wikipedia focuses on using only a single language. The main contribution of this paper is that by using a huge multilingual data resource (in our case Wikipedia) combined with a structured monolingual resource such as WordNet, we have shown that it is possible to extend a monolingual resource to other languages. Our results show that the method is quite consistent and effective for this task.

The same experiment can be repeated using Wikipedia and WordNet on languages others than Japanese and Spanish offering useful results especially for minority languages.

In addition, the use of Wikipedia and WordNet in combination achieves better results than those that could be achieved using either resource independently.

We plan to extent the coverage of the thesaurus to other syntactic categories such as verbs, adverb, and adjectives. We also evaluate our thesaurus in real world tasks such as the construction of comparable corpora for use in MT.

Acknowledgments

We would like to thanks to Eric Nichols for his helpful comments.

References

K. Ahmad, M. Tariq, B. Vrusias and C. Handy. 2003. Corpus-Based Thesaurus Construction for Image Retrieval in Specialist Domains. In *Proceedings of* ECIR 2003. pp. 502-510.

R. Bunescu and M. Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of* EACL-06, pp. 9-16.

C. Fellbaum. 1998. WordNet: *An Electronic Lexical Database*. Cambridge, Mass.: MIT press. pp. 25-43.

J. Honglan and Kam-Fai-Won. 2002. A Chinese dictionary construction algorithm for information retrieval. ACM Transactions on Asian Language Information Processing. pp. 281-296.

C. Manning and H. Schütze. 2000. Foundations of Statistical Natural Language Processing. Cambridge, Mass.: MIT press. pp. 230-259.

K. Oi Yee, 1998. Bridging the Gap between Dictionary and Thesaurus. COLING-ACL. pp. 1487-1489.

R. Rada, H. Mili, E. Bicknell and M. Blettner. 1989. Development and application of a metric semantic nets. IEEE Transactions on Systems, Man and Cybernetics, 19(1):17-30.

M. Ruiz, E. Alfonseca and P. Castells. 2005. Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets. In *Proceedings of* AWIC-05. Lecture Notes in Computer Science 3528. pp. 380-386, Springer, 2005.

M. Strube and S. P. Ponzetto. 2006. WikiRelate! Computing semantic relatedness using Wikipedia. 21st National Conference on Artificial Intelligence.

L. Urdang. 1991. The Oxford Thesaurus. Clarendon press. Oxford.

Automatic Identification of Rhetorical Roles using Conditional Random Fields for Legal Document Summarization

M. Saravanan
Department of CS & E
IIT Madras, Chennai-36
msdess@yahoo.com

B. Ravindran
Department of CS & E
IIT Madras, Chennai-36
ravi@cse.iitm.ac.in

S. Raman
Department of CS & E
IIT Madras, Chennai-36
ramansubra@gmail.com

Abstract

In this paper, we propose a machine learning approach to rhetorical role identification from legal documents. In our approach, we annotate roles in sample documents with the help of legal experts and take them as training data. Conditional random field model has been trained with the data to perform rhetorical role identification with reinforcement of rich feature sets. The understanding of structure of a legal document and the application of mathematical model can brings out an effective summary in the final stage. Other important new findings in this work include that the training of a model for one sub-domain can be extended to another sub-domains with very limited augmentation of feature sets. Moreover, we can significantly improve extraction-based summarization results by modifying the ranking of sentences with the importance of specific roles.

1 Introduction

With the availability of large number of colossal legal documents in electronic format, there is a rising need for effective information retrieval tools to assist in organizing, processing and retrieving this information and presenting them in a suitable user-friendly format. To that end, text summarization is an important step for many of these larger information management goals. In recent years, much attention has been focused on the problem of understanding the structure and textual units in legal judgments (Farzindar & Lapalme, 2004). In this case, performing automatic segmentation of a document to understand the rhetorical roles turns out to be an important research issue. For instance, Farzindar (2004) proposed a text summarization method to manipulate factual and heuristic knowledge from legal documents. Hachey and Grover (2005) explored machine learning approach to rhetorical status classification by performing fact extraction and sentence extraction for automatic summarization of texts in the legal domain. They formalized the problem to extract most important units based on the identification of thematic structure of the document and determination of argumentative roles of the textual units in the judgment. They mainly used linguistic features to identify the thematic structures.

In this paper, we discuss methods for automatic identification of rhetorical roles in legal judgments based on rules and on machine learning techniques. Using manually annotated sample documents on three different legal sub-domains (rent control, income tax and sales tax), we train an undirected graphical model to segment the documents along different rhetorical structures. To represent the documents for this work, we mainly used features like cue words, state transition, named entity, position and other local and global features. The segmented texts with identified roles play a crucial part in re-ordering the ranking in the final extraction-based summary. The important sentences are extracted based on the term distribution model given in [Saravanan *et al*, 2006]. In order to develop a generic approach to perform segmentation, we use a fixed set of seven rhetorical categories based on Bhatia's (1993) genre analysis shown in Table 1.

Graphical Models are nowadays used in many text processing applications; however the main

Rhetorical Roles	*Description*
Identifying the case (1)	The sentences that are present in a judgment to identify the issues to be decided for a case. Courts call them as "Framing the issues".
Establishing facts of the case (2)	The facts that are relevant to the present proceedings/litigations that stand proved, disproved or unproved for proper applications of correct legal principle/law.
Arguing the case (3)	Application of legal principle/law advocated by contending parties to a given set of proved facts.
History of the case (4)	Chronology of events with factual details that led to the present case between parties named therein before the court on which the judgment is delivered.
Arguments (Analysis) (5)	The court discussion on the law that is applicable to the set of proved facts by weighing the arguments of contending parties with reference to the statute and precedents that are available.
Ratio decidendi (6) *(Ratio of the decision)*	Applying the correct law to a set of facts is the duty of any court. The reason given for application of any legal principle/law to decide a case is called Ratio decidendi in legal parlance. It can also be described as the central generic reference of text.
Final decision (7) *(Disposal)*	It is an ultimate decision or conclusion of the court following as a natural or logical outcome of ratio of the decision

Table 1. The current working version of the rhetorical annotation scheme for legal judgments.

focus has been performing Natural Language processing tasks on newspaper and research paper domains. As a novel approach, we have tried and implemented the CRF model for role identification in legal domain. In this regard, we have first implemented rule based approach and extend this method with additional features and a probabilistic model. In another study, CRF is used as a tool to model the sequence labeling problem for summarization task (Shen at al., 2006). In our work, we are in the process of developing a fully automatic summarization system for a legal domain on the basis of Lafferty's (2001) segmentation task and Teufel & Moen's (2004) gold standard approaches. Legal judgments are different in characteristics compared with articles reporting scientific research papers and other simple domains related to the identification of basic structures of a document. To perform a summarization methodology and find out important portions of a legal document is a complex problem (Moens, 2004). Even the skilled lawyers are facing difficulty in identifying the main decision part of a law report. The genre structure identified for legal judgment in our work plays a crucial role in identifying the main decision part in the way of breaking the document in anaphoric chains. The sentence extraction task forms part of an automatic summarization system in the legal domain. The main focus of this paper is information extraction task based on the identified roles and methods of structuring summaries which has considered being a hot research topic (Yeh *et al*., 2005). Now we will discuss the importance of identifying rules in the data collection by various methods available for rule learning in the next section.

2 Text Segmentation Algorithms

We explain two approaches to text segmentation for identifying the rhetorical roles in legal judgments. The focus of the first approach is on a rule-based method with novel rule sets which we fine-tuned for legal domains. That is, we frame text segmentation as a rule learning problem. The proposed rule-based method can be enhanced with additional features and a probabilistic model. An undirected graphical model, Conditional Random Field (CRF) is used for this purpose. It shows significant improvement over the rule-based method. The explanation of these methods is given in the following sections.

2.1 Rule-based learning algorithms

Most traditional rule learning algorithms are based on a divide-and-conquer strategy. SLIPPER [Cohen, 1999] is one of the standard rule learning algorithms used for information retrieval task. In SLIPPER, the ad hoc metrics used to guide the growing and pruning of rules are replaced with metrics based on the formal analysis of boosting algorithms. For each instance, we need to check each and every rule in the rule set for a given sentence. It takes more time for larger corpora

compared to other rule learning algorithms even for a two-class problem. If we need to consider more than two classes and to avoid overfitting of ensemble of rules, one has to think of grouping the rules in a rule set and some chaining mechanism has to be followed. Another rule learning algorithm RuleFit (Friedman & Popescu, 2005) generates a small comprehensible rule set which is used in ensemble learning with larger margin. In this case, overfitting may happen, if the rule set gets too large and thus some form of control has to be maintained. Our main idea is to find a preferably small set of rules with high predictive accuracy and with marginal execution time.

We propose an alternative rule learning strategy that concentrates on classification of rules and chaining relation in each rhetorical role (Table 1) based on the human annotation schemes. A chain relation is a technique used to identify co-occurrences of roles in legal judgments. In our approach, rules are conjunctions of primitive conditions. As used by the boosting algorithms, a rule set R can be any hypothesis that partitions the set of instance X into particular role categorization; the set of instances which satisfy any one of seven different set of categorized roles. We start by generating rules that describe the original features found in the training set. Each rule outputs 1 if its condition is met, 0 if it is not met. Let us now define for a sample document X = (S1, S2,….,Sm) of size m, we assume that the set of rules $R = \{r_1, r_2, \ldots\}$ are applied to sample X, where each rule $r_i : X \rightarrow L$ represents the mapping of sentences of X onto a rhetorical role and $L = \{L_1, L_2, \ldots, L_7\}$. Each L_i represents a rhetorical role from the fixed set shown in Table 1. An outline of our method is given below.

```
Procedure Test (X)
 {   Read test set
     Read instances from sample X (instances may be
      words, N-grams or even full sentences)
      Apply rules in R (with role categorization
             by maintaining chain relation)
      For k = 1 to m sentences
        For i = 1, 2, …. no. of instances in each sentence
        For j = 1 to 7      /* 7 identified roles */
        If there exist a rule which satisfies then
             X(i,j) gets a value 1
        Else
          X(i,j) gets a value {1,0} based on chain relation
        S(k) = L (argmax Σ(X(i,j)))
                    j   i
}
```

2.2 Conditional Random Fields and Features

The CRF model-based retrieval system designed in this paper will depict the way a human can summarize a legal judgment by understanding the importance of roles and related contents. Conditional Random Fields is one of the recently emerging graphical models which have been used for text segmentation problems and proved to be one of the best available frame works compared to other existing models (Lafferty, 2001). A judgment can be regarded as a sequence of sentences that can be segmented along the seven rhetorical roles where each segments is relatively coherent in content. We use CRF as a tool to model the text segmentation problem. CRFs are undirected graphical models used to specify the conditional probabilities of possible label sequences given an observation sequence. Moreover, the conditional probabilities of label sequences can depend on arbitrary, non independent features of the observation sequence, since we are not forming the model to consider the distribution of those dependencies. In a special case in which the output nodes of the graphical model are linked by edges in a linear chain, CRFs make a first-order Markov independence assumption with binary feature functions, and thus can be understood as conditionally-trained finite state machines (FSMs) which are suitable for sequence labeling.

A linear chain CRF with parameters $C = \{C_1, C_2, \ldots..\}$ defines a conditional probability for a label sequence $l = l_1, \ldots..l_w$ (e.g., Establishing facts of the case, Final decision, etc.) given an observed input sequence $s = s_1, \ldots s_W$ to be

$$P_C(l \mid s) = \frac{1}{Z_s} \exp\left[\sum_{t=1}^{w} \sum_{k=1}^{m} C_k f_k(l_{t-1}, l_t, s, t)\right] \quad \ldots. \quad (1)$$

where Z_s is the normalization factor that makes the probability of all state sequences sum to one, $f_k(l_{t-1}, l_t, s, t)$ is one of m feature functions which is generally binary valued and C_k is a learned weight associated with feature function. For example, a feature may have the value of 0 in most cases, but given the text "points for consideration", it has the value 1 along the transition where l_{t-1} corresponds to a state with the label ***identifying the case,*** l_t corresponds to a state with the label ***history of the case,*** and f_k is the feature function PHRASE=

"points for consideration" belongs to s at position t in the sequence. Large positive values for C_k indicate a preference for such an event, while large negative values make the event unlikely and near zero for relatively uninformative features. These weights are set to maximize the conditional log likelihood of labeled sequence in a training set D = {(s_t, l_t) : t = 1,2,…w), written as:

$$L_C(D) = \sum_i \log P_C(l_i \mid s_i)$$

$$= \sum_i \left(\sum_{t=1}^{w} \sum_{k=1}^{m} C_k f_k(l_{t-1}, l_t, s, t) - \log Z s_i \right) \ldots (2)$$

The training state sequences are fully labeled and definite, the objective function is convex, and thus the model is guaranteed to find the optimal weight settings in terms of L_C (D). The probable labeling sequence for an input s_i can be efficiently calculated by dynamic programming using modified Viterbi algorithm. These implementations of CRFs are done using newly developed java classes which also use a quasi-Newton method called L-BFGS to find these feature weights efficiently. In addition to the following standard set of features, we also added other related features to reduce the complexity of legal domain.

Indicator/cue phrases – The term 'cue phrase' indicates the key phrases frequently used which are the indicators of common rhetorical roles of the sentences (e.g. phrases such as "We agree with court", "Question for consideration is", etc.,). In this study, we encoded this information and generated automatically explicit linguistic features. Feature functions for the rules are set to 1 if they match words/phrases in the input sequence exactly.

Named entity recognition - This type of recognition is not considered fully in summarizing scientific articles (Teufel & Moens, 2002). But in our work, we included few named entities like Supreme Court, Lower court etc., and generate binary-valued entity type features which take the value 0 or 1 indicating the presence or absence of a particular entity type in the sentences.

Local features and Layout features - One of the main advantages of CRFs is that they easily afford the use of arbitrary features of the input. One can encode abbreviated features; layout features such as position of paragraph beginning, as well as the sentences appearing with quotes, all in one framework.

State Transition features - In CRFs, state transitions are also represented as features (Peng & McCullam, 2006). The feature function $f_k(l_{t-1}, l_t, s, t)$ in Eq. (1) is a general function over states and observations. Different state transition features can be defined to form different Markov-order structures. We define state transition features corresponding to appearance of years attached with Section and Act nos. related to the labels *arguing the case* and *arguments*.

Legal vocabulary features - One of the simplest and most obvious set of features is decided using the basic vocabularies from a training data. The words that appear with capitalizations, affixes, and in abbreviated texts are considered as important features. Some of the phrases that include **v.** and *act/section* are the salient features for *arguing the case* and *arguments* categories.

2.3 Experiments with role identification

We have gathered a corpus of legal judgments up to the year 2006 which were downloaded from www.keralawyer.com specific to the sub-domains of rent control, income tax and sales tax. Using the manually annotated subset of the corpus (200 judgments) we have performed a number of preliminary experiments to determine which method would be appropriate for role identification. The annotated corpus is available from iil.cs.iitm.ernet.in/datasets. Even though, income tax and sales tax judgments are based on similar facts, the number of relevant legal sections / provisions are differ. The details and structure of judgments related to rent control domain are not the same compared to income tax and sales tax domains. Moreover, the roles like ratio decidendi and final decision occur many times spread over the full judgment in sales tax domain, which is comparatively different to other sub-domains. We have implemented both the approaches on rent control domain successfully. We found that the other sub-domains need specific add-on features which improve the result by an additional 20%. Based on this, we have introduced additional features and new set of rules for the income tax and sales tax related judgments. The modified rule set and additional features are smaller in number, but create a good impact on the rhetorical status

	Rhetorical Roles	Precision			Recall			F-measure		
		Slipper	Rule-based	CRF	Slipper	Rule-based	CRF	Slipper	Rule-based	CRF
Rent Control Domain	Identifying the case	0.641	0.742	0.846	0.512	0.703	0.768	0.569	0.722	0.853
	Establishing the facts of the case	0.562	0.737	0.824	0.456	0.664	0.786	0.503	0.699	0.824
	Arguing the case	0.436	0.654	0.824	0.408	0.654	0.786	0.422	0.654	0.805
	History of the case	0.841	0.768	0.838	0.594	0.716	0.793	0.696	0.741	0.815
	Arguments	0.543	0.692	0.760	0.313	0.702	0.816	0.397	0.697	0.787
	Ratio of decidendi	0.574	0.821	0.874	0.480	0.857	0.903	0.523	0.839	0.888
	Final Decision	0.700	0.896	0.986	0.594	0.927	0.961	0.643	0.911	0.973
							Micro-Average of F-measure	**0.536**	**0.752**	**0.849**
	Rhetorical Roles	Precision			Recall			F-measure		
		Slipper	Rule-based	CRF	Slipper	Rule-based	CRF	Slipper	Rule-based	CRF
Income Tax Domain	Identifying the case	0.590	0.726	0.912	0.431	0.690	0.852	0.498	0.708	0.881
	Establishing the facts of the case	0.597	0.711	0.864	0.512	0.659	0.813	0.551	0.684	0.838
	Arguing the case	0.614	0.658	0.784	0.551	0.616	0.682	0.581	0.636	0.729
	History of the case	0.437	0.729	0.812	0.418	0.724	0.762	0.427	0.726	0.786
	Arguments	0.740	0.638	0.736	0.216	0.599	0.718	0.334	0.618	0.727
	Ratio of decidendi	0.416	0.708	0.906	0.339	0.663	0.878	0.374	0.685	0.892
	Final Decision	0.382	0.752	0.938	0.375	0.733	0.802	0.378	0.742	0.865
							Micro-Average of F-measure	**0.449**	**0.686**	**0.817**
	Rhetorical Roles	Precision			Recall			F-measure		
		Slipper	Rule-based	CRF	Slipper	Rule-based	CRF	Slipper	Rule-based	CRF
Sales Tax Domain	Identifying the case	0.539	0.675	0.842	0.398	0.610	0.782	0.458	0.641	0.811
	Establishing the facts of the case	0.416	0.635	0.784	0.319	0.559	0.753	0.361	0.595	0.768
	Arguing the case	0.476	0.718	0.821	0.343	0.636	0.747	0.399	0.675	0.782
	History of the case	0.624	0.788	0.867	0.412	0.684	0.782	0.496	0.732	0.822
	Arguments	0.500	0.638	0.736	0.438	0.614	0.692	0.467	0.626	0.713
	Ratio of decidendi	0.456	0.646	0.792	0.318	0.553	0.828	0.375	0.596	0.810
	Final Decision	0.300	0.614	0.818	0.281	0.582	0.786	0.290	0.598	0.802
							Micro-Average of F-measure	**0.407**	**0.637**	**0.787**

Table 2. Precision, Recall and F-measure for seven rhetorical roles

classification in the sales tax and income tax domains. It is common practice to consider human performances as an upper bound for most of the IR tasks, so in our evaluation, the performance of the system has been successfully tested by matching with human annotated documents.

Kappa (Siegal & Castellan, 1988) is an evaluation measure used in our work to compare the inter-agreement between sentences extracted by two human annotators for role identification in legal judgments. The value (K=0.803) shows the good reliability of human annotated corpus. The results given in Table 2 show that CRF-based and rule-based methods perform well for each role categories compared to SLIPPER method. CRF-based method performs extremely well and paired t-test result indicates that it is significantly ($p < .01$) higher than the other two methods on rhetorical role identification for legal judgments belonging to rent control, income tax and sales tax sub-domains. In this experiment, we also made an effort to understand the annotation of relevance of seven rhetorical categories.

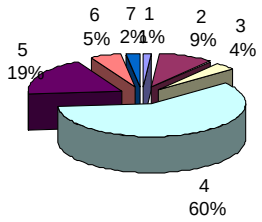


Figure 1. Distribution of rhetorical roles (10 entire documents from rent control sub-domain)

Figure 1 shows that the distribution of the seven categories is very much skewed, with 60% of all sentences being classified as *history of the case*. Basically it includes the remaining contents of the

document other than the six categories. In this case, we have calculated the distribution among 10 judgments related to rent control documents. Figure 2 shows the rhetorical category distribution among the 10 different summaries from rent control domain. This shows that the resulting category distribution is far more evenly distributed than the one covering all sentences in Figure 1. *Ratio of decidendi* and *final decision* are the two most frequent categories in the sentences extracted from judgments. The label numbers mentioned in the Figures denote the rhetorical roles which as defined in Table 1.

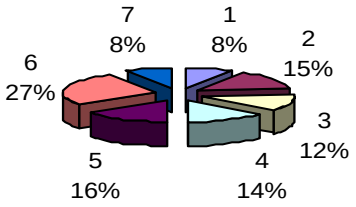


Figure 2. Distribution of rhetorical roles (10 different summaries from rent control sub-domain)

3 Legal Document Summarization

Extraction of sentences in the generation of a summary at different percentage levels of text is one of the widely used methods in document summarization (Radev *et al.*, 2002). For the legal domain, generating a summary from the original judgment is a complex problem. Our approach to produce the summary is extraction-based method which identifies important elements present in a legal judgment. The identification of the document structure using CRF-model categorizes the key ideas from the details of a legal judgment. The genre structure has been applied to final summary to improve the readability and coherence. In order to evaluate the effectiveness of our summarizer, we have applied four different measures to look for a match on the model summary generated by humans (*head notes*) from the text of the original judgments.

3.1 Applying term distribution model

The automatic text summarization process starts with sending legal document to a preprocessing stage. In this preprocessing stage, the document is to be divided into segments, sentences and tokens. We have introduced some new feature identification techniques to explore paragraph alignments. This process includes the understanding of abbreviated texts and section numbers and arguments which are very specific to the structure of legal documents. The other useful statistical natural language processing tools, such as filtering out stop list words, stemming etc., are carried out in the preprocessing stage. The resulting intelligible words are useful in the normalization of terms in the term distribution model (Saravanan *et al.*, 2006). During the final stage, we have altered the ranks or removed some of the sentences from the final summary based on the structure discovered using CRF. The summarization module architecture is shown in Figure 3.

The application of term distribution model brings out a good extract of sentences present in a legal document to generate a summary. The sentences with labels identified during CRF implementation can be used with the term distribution model to give more significance to some of the sentences with specific roles. Moreover, the structure details available in this stage are useful in improving the coherency and readability among the sentences present in the summary.

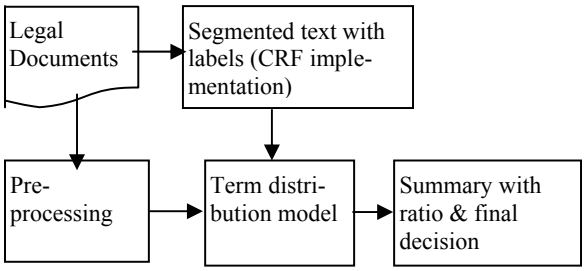


Figure 3. Architectural view of summarization system.

3.2 Evaluation of a summary

Extrinsic and intrinsic are the two different evaluation strategies available for text summarization (Sparck Jones & Gablier, 1996). Intrinsic measure shows the presence of source contents in the summary. F-measure and MAP are two standard intrinsic measures used for the evaluation of our system-generated summary. We have also used ROUGE evaluation approach (Lin, 2004) which is based on n-gram co-occurrences between machine summaries and *ideal* human summaries.

In this paper, we have applied ROUGE-1 and ROUGE-2 which are simple n-gram measures. We compared our results with Microsoft, Mead Summarizer (Radev et al., 2003) and other two simple baselines: one which chooses 15% of words of the beginning of the judgment and second chooses last 10% of words of the judgment with human reference summaries. Both the baselines defined in this study are standard baselines for newspaper and research domains. The result shown in Table 3 highlights the better performances of our summarizer compared to other methods considered in this study. We can see that the results of MEAD and WORD summaries are not at the expected level, while our summarizer is best in terms of all four evaluation measures. Results are clearly indicated that our system performs significantly better than the other systems for legal judgments.

	MAP	F-measure	ROUGE-1	ROUGE-2
Baseline 1	0.370	0.426	0.522	0.286
Baseline 2	0.452	0.415	0.402	0.213
Microsoft Word	0.294	0.309	0.347	0.201
Mead	0.518	0.494	0.491	0.263
Our system	**0.646**	**0.654**	**0.685**	**0.418**

Table 3. MAP, F-measure and ROUGE scores.

4 Conclusion

This paper describes a novel method for generating a summary for legal judgments with the help of undirected graphical models. We observed that rhetorical role identification from legal documents is one of the primary tasks to understand the structure of the judgments. CRF model performs much better than rule based and other rule learning method in segmenting the text for legal domains. Our approach to summary extraction is based on the extended version of term weighting method. With the identified roles, the important sentences generated in the probabilistic model will be reordered or suppressed in the final summary. The evaluation results show that the summary generated by our summarizer is closer to the human generated head notes, compared to the other methods considered in this study. Hence the legal community will get a better insight without reading a full judgment. Moreover, our system-generated summary is more useful for lawyers to prepare the case history related to presently appearing cases.

Acknowledgement

We would like to thank the legal fraternity for the assistance and guidance governs to us. Especially we express our sincere gratitude to the advocates Mr. S.B.C. Karunakaran and Mr. K.N. Somasundaram for their domain advice and continuous guidance in understanding the structure of legal document and for hand annotated legal judgments.

References

Atefeh Farzindar and Guy Lapalme. 2004. *Legal text summarization by exploration of the thematic structures and argumentative roles,* In Text summarization Branches out workshop held in conjunction with ACL 2004, pages 27-34, Barcelona, Spain.

Atefeh Farzindar and Guy Lapalme. 2004. *Letsum, an automatic legal text summarizing system*, Legal Knowledge and Information System, Jurix 2004: The Seventeenth Annual Conference, Amsterdam, IOS Press, PP.11-18.

Ben Hachey and Claire Grover. 2005. *Sequence Modeling for sentence classification in a legal summarization system*, Proceedings of the 2005 ACM symposium on Applied Computing.

Bhatia, V.K., 1999. *Analyzing Genre: Language Use in Professional Settings*, London, Longman.

Cohen,W., and Singer, Y. 1999. *A simple, fast, and effective rule learner*, Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), AAAI Press, pp.335-342.

Dragomir Radev, Eduard Hovy, Kathleen McKeown. 2002. *Introduction to the special issue on summarization,* Computational Linguistics 28(4)4, Association for Computing Machinery.

Dragomir Radev, Jahna Otterbaher, Hong Qi, and Daniel Tam. 2003. *Mead Reducs: Michigan at DUC, 2003.* In DUC03, Edmonton, Alberta, Canada, May 31- June 1. Association for Computational Linguistics.

Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. *Document Summarization using Conditional Random Fields.* International Joint Conference on Artificial Intelligence, IJCAI 2007, Hyderabad, India, PP.2862-2867.

Friedmen, J.H., & and Popescu, B. E. 2005. *Predictive learning via rule ensembles* (Technical Report), Stanford University.

Fuchun Peng and Andrew McCullam, 2006. *Accurate information extraction from research papers using conditional random fields*, Information Processing Management, 42(4): 963-979.

John Lafferty, Andrew McCullam and Fernando Pereira, 2001. *Conditional Random Fields: Probabilistic models and for segmenting and labeling sequence data*, Proceedings of international conference on Machine learning.

Karen Sparck Jones and Julia Galliers. 1996. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Natural Language Engineering, 4(2):175–190, Springer-Verlag.

Lin, Chin-Yew. 2004. *ROUGE: a Package for Automatic Evaluation of Summaries,* Proceedings of Workshop on Text Summarization, pp: 21--26, Barcelona, Spain.

Marie-Francine Moens, 2004. *An Evaluation Forum for Legal Information Retrieval Systems*? Proceedings of the ICAIL-2003 Workshop on Evaluation of Legal Reasoning and Problem-Solving Systems (pp. 18-24). International Organization for Artificial Intelligence and Law.

Saravanan , M., Ravindran, B. and Raman, S. 2006. *A Probabilistic Approach to Multi-document summarization for generating a Tiled Sumamry*, International Journal of Computational Intelligence and Applications, 6(2): 231-243, Imperial College Press.

Saravanan , M., Ravindran, B. and Raman, S. 2006. *Improving legal document Summarization using graphical models*, Legal Knowledge and Information System, JURIX 2006: The Nineteenth Annual Conference, Paris, IOS Press, PP.51-60.

Siegal, Sidney and N.John Jr. Castellan. 1988. *Non-parametric statistics for the behavioral sciences*, McGraw Hill, Berkeley, CA.

Simone Teufel and Marc Moens, 2002. *Summarizing scientific articles – experiments with relevance and rhetorical status,* Association of Computational Linguistics, 28(4): 409-445.

Yen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng, 2005. *Text summarization using a trainable summarizer and latent semantic analysis*, Information processing management, 41(1):75-95.

Projection-based Acquisition of a Temporal Labeller

Kathrin Spreyer*
Department of Linguistics
University of Potsdam
Germany
spreyer@uni-potsdam.de

Anette Frank
Dept. of Computational Linguistics
University of Heidelberg
Germany
frank@cl.uni-heidelberg.de

Abstract

We present a cross-lingual projection framework for temporal annotations. Automatically obtained TimeML annotations in the English portion of a parallel corpus are transferred to the German translation along a word alignment. Direct projection augmented with shallow heuristic knowledge outperforms the uninformed baseline by 6.64% F_1-measure for events, and by 17.93% for time expressions. Subsequent training of statistical classifiers on the (imperfect) projected annotations significantly boosts precision by up to 31% to 83.95% and 89.52%, respectively.

1 Introduction

In recent years, supervised machine learning has become the standard approach to obtain robust and wide-coverage NLP tools. But manually annotated training data is a scarce and expensive resource. *Annotation projection* (Yarowsky and Ngai, 2001) aims at overcoming this resource bottleneck by scaling conceptually monolingual resources and tools to a multilingual level: annotations in existing monolingual corpora are transferred to a different language along the word alignment to a parallel corpus.

In this paper, we present a projection framework for *temporal annotations*. The TimeML specification language (Pustejovsky et al., 2003a) defines an annotation scheme for time expressions (*timex* for short) and events, and there are tools for the automatic TimeML annotation of English text (Verhagen et al., 2005). Similar rule-based systems exist for Spanish and Italian (Saquete et al., 2006). However, such resources are restricted to a handful of languages.

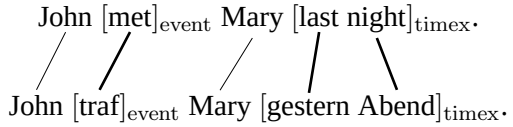


Figure 1: Annotation projection.

We employ the existing TimeML labellers to annotate the English portion of a parallel corpus, and automatically project the annotations to the word-aligned German translation. Fig. 1 shows a simple example. The English sentence contains an event and a timex annotation. The event-denoting verb *met* is aligned with the German *traf*, hence the latter also receives the event tag. Likewise, the components of the multi-word timex *last night* align with German *gestern* and *abend*, respectively, and the timex tag is transferred to the expression *gestern abend*.

Projection-based approaches to multilingual annotation have proven adequate in various domains, including part-of-speech tagging (Yarowsky and Ngai, 2001), NP-bracketing (Yarowsky et al., 2001), dependency analysis (Hwa et al., 2005), and role semantic analysis (Padó and Lapata, 2006). To our knowledge, the present proposal is the first to apply projection algorithms to temporal annotations.

* The first author was affiliated with Saarland University (Saarbrücken, Germany) at the time of writing.

Cross-lingually projected information is typically noisy, due to errors in the source annotations as well as in the word alignment. Moreover, successful projection relies on the *direct correspondence assumption* (*DCA*, Hwa et al. (2002)) which demands that the annotations in the source text be homomorphous with those in its (literal) translation. The DCA has been found to hold, to a substantial degree, for the above mentioned domains. The results we report here show that it can also be confirmed for temporal annotations in English and German. Yet, we cannot preclude *divergence* from translational correspondence; on the contrary, it occurs routinely and to a certain extent systematically (Dorr, 1994). We employ two different techniques to filter noise. Firstly, the projection process is equipped with (partly language-specific) knowledge for a principled account of typical alignment errors and cross-language discrepancies in the realisation of events and timexes (section 3.2). Secondly, we apply aggressive data engineering techniques to the noisy projections and use them to train statistical classifiers which generalise beyond the noise (section 5).

The paper is structured as follows. Section 2 gives an overview of the TimeML specification language and compatible annotation tools. Section 3 presents our projection models for temporal annotations, which are evaluated in section 4. Section 5 describes how we induce temporal labellers for German from the projected annotations; section 6 concludes.

2 Temporal Annotation

2.1 The TimeML Specification Language

The TimeML specification language (Pustejovsky et al., 2003a)[1] and annotation framework emerged from the TERQAS workshop[2] in the context of the ARDA AQUAINT programme. The goal of the programme is the development of question answering (QA) systems which index content rather than plain keywords. Semantic indexing based on the identification of named entities in free text is an established method in QA and related applications. Recent years have also seen advances in relation extraction, a variant of event identification, albeit restricted in terms of coverage: the majority of systems addressing the task use a pre-defined set of—typically domain-specific—templates. In contrast, TimeML models events in a domain-independent manner and provides principled definitions for various event classes. Besides the identification of *events*, it addresses their relative ordering and anchoring in time by integrating *timexes* in the annotation. The major contribution of TimeML is the explicit representation of dependencies (so-called *links*) between timexes and events.

Unlike traditional accounts of events (e.g., Vendler (1967)), TimeML adopts a very broad notion of eventualities as "situations that happen or occur" and "states or circumstances in which something obtains or holds true" (Pustejovsky et al., 2003a); besides verbs, this definition includes event nominals such as *accident*, and stative modifiers (*prepared*, *on board*). Events are annotated with `EVENT` tags. TimeML postulates seven event classes: REPORTING, PERCEPTION, ASPECTUAL, I-ACTION, I-STATE, STATE, and OCCURRENCE. For definitions of the individual classes, the reader is referred to Saurí et al. (2005b).

Explicit timexes are marked by the `TIMEX3` tag. It is modelled on the basis of Setzer's (2001) `TIMEX` tag and the TIDES `TIMEX2` annotation (Ferro et al., 2005). Timexes are classified into four types: dates, times, durations, and sets.

Events and timexes are interrelated by three kinds of links: temporal, aspectual, and subordinating. Here, we consider only *subordinating links (slinks)*. Slinks explicate event modalities, which are of crucial importance when reasoning about the certainty and factuality of propositions conveyed by event-denoting expressions; they are thus directly relevant to QA and information extraction applications. Slinks relate events in modal, factive, counterfactive, evidential, negative evidential, or conditional relationships, and can be triggered by lexical or structural cues.

2.2 Automatic Labellers for English

The basis of any projection architecture are high-quality annotations of the source (English) portion

[1] A standardised version ISO-TimeML is in preparation, cf. Schiffrin and Bunt (2006).

[2] See `http://www.timeml.org/site/terqas/index.html`

$e \in E$	temporal entity
$l \in E \times E$	(subordination) link
$w_s \in W_s, w_t \in W_t$	source/target words
$al \in Al : W_s \times W_t$	word alignment
$A_s \ni a_s : E \rightarrow 2^{W_s}$	source annotation
$A_t \ni a_t :$	projected target
$(E \times A_s \times Al) \rightarrow 2^{W_t}$	annotation

Table 1: Notational conventions.

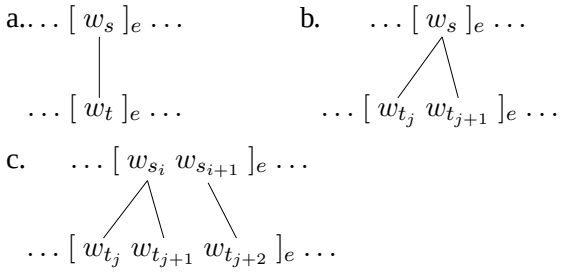


Figure 2: Projection scenarios: (a) single-word 1-to-1, (b) single-word 1-to-many, (c) multi-word.

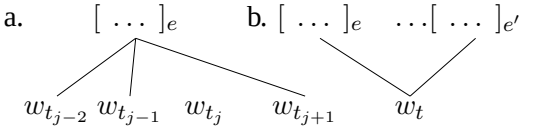


Figure 3: Problematic projection scenarios: (a) non-contiguous aligned span, (b) rivalling tags.

of the parallel corpus. However, given that the projected annotations are to provide enough data for training a target language labeller (section 5), manual annotation is not an option. Instead, we use the TARSQI tools for automatic TimeML annotation of English text (Verhagen et al., 2005). They have been modelled and evaluated on the basis of the TimeBank (Pustejovsky et al., 2003b), yet for the most part rely on hand-crafted rules. To obtain a full temporal annotation, the modules are combined in a cascade. We are using the components for timex recognition and normalisation (Mani and Wilson, 2000), event extraction (Saurí et al., 2005a), and identification of modal contexts (Saurí et al., 2006).[3]

3 Informed Projection

3.1 The Core Algorithm

Recall that TimeML represents temporal entities with `EVENT` and `TIMEX3` tags which are anchored to words in the text. Slinks, on the other hand, are not anchored in the text directly, but rather relate temporal entities. The projection of links is therefore entirely determined by the projection of the entities they are defined on (see Table 1 for the notation used throughout this paper): a link $l = (e, e')$ in the source annotation a_s projects to the target annotation a_t iff both e and e' project to non-empty sequences of words. The projection of the entities e, e' themselves, however, is a non-trivial task.

[3]TARSQI also comprises a component that introduces temporal links (Mani et al., 2003); we are not using it here because the output includes the entire tlink closure. Although Mani et al. (2006) use the links introduced by closure to boost the amount of training data for a tlink classifier, this technique is not suitable for our learning task since the closure might easily propagate errors in the automatic annotations.

Given a temporal entity e covering a sequence $a_s(e)$ of tokens in the source annotation, the projection model needs to determine the extent $a_t(e, a_s, al)$ of e in the target annotation, based on the word alignment al. Possible projection scenarios are depicted in Fig. 2. In the simplest case (Fig. 2a), e spans a single word w_s which aligns with exactly one word w_t in the target sentence. In this case, the model predicts e to project to w_t. A single tagged word with 1-to-many alignments (as in Fig. 2b) requires a more thorough inspection of the aligned words. If they form a contiguous sequence, e can be projected onto the entire sequence as a multi-word unit. This is problematic in a scenario such as the one shown in Fig. 3a, where the aligned words do *not* form a contiguous sequence. There are various strategies, described in section 3.2, to deal with non-contiguous cases. For the moment, we can adopt a conservative approach which categorically blocks discontinuous projections. Finally, Fig. 2c illustrates the projection of an entity spanning multiple words. Here, the model composes the projection span of e from the alignment contribution of each individual word w_s covered by e. Again, the final extent of the projected entity is required to be contiguous.

With any of these scenarios, a problem arises when two distinct entities e and e' in the source an-

1. project(a_s, al):
2. $a_{t,\mathsf{C}} = \emptyset$
3. for each entity e defined by a_s:
4. $a_{t,\mathsf{C}}(e, a_s, al) = \bigcup^{\mathsf{C}}_{w_s \in a_s(e)} \mathsf{proj}(w_s, e, a_s, al)$
5. for each link $l = (e, e')$ defined over a_s:
6. if $a_{t,\mathsf{C}}(e, a_s, al) \neq \emptyset$ and $a_{t,\mathsf{C}}(e', a_s, al) \neq \emptyset$
7. then define l to hold for $a_{t,\mathsf{C}}$
8. return $a_{t,\mathsf{C}}$

where

$$\mathsf{proj}(w_s, e, a_s, al) = \{w_t \in W_t \mid (w_s, w_t) \in al \wedge \forall e' \in a_s.\ e' \neq e \Rightarrow w_t \notin a_{t,\mathsf{C}}(e', a_s, al)\}$$

and

$$\bigcup^{\mathsf{C}} S = \begin{cases} \bigcup S & : \bigcup S \text{ is convex} \\ \emptyset & : \text{otherwise} \end{cases}$$

Figure 4: The projection algorithm.

notation have conflicting projection extents, that is, when $a_t(e, a_s, al) \cap a_t(e', a_s, al) \neq \emptyset$. This is illustrated in Fig. 3b. The easiest strategy to resolve conflicts like these is to pick an arbitrary entity and privilege it for projection to the target word(s) w_t in question. All other rivalling entities e' project onto their remaining target words $a_t(e', a_s, al) \setminus \{w_t\}$.

Pseudocode for this word-based projection of temporal annotations is provided in Fig. 4.

3.2 Incorporating Additional Knowledge

The projection model described so far is extremely susceptible to errors in the word alignment. Related efforts (Hwa et al., 2005; Padó and Lapata, 2006) have already suggested that additional linguistic information can have considerable impact on the quality of the projected annotations. We therefore augment the baseline model with several shallow heuristics encoding linguistic or else topological constraints for the choice of words to project to. Linguistically motivated filters refer to the part-of-speech (POS) tags of words in the target language sentence, whereas topological criteria investigate the alignment topology.

Linguistic constraints. Following Padó and Lapata (2006), we implement a filter which discards alignments to non-content words, for two reasons: (i) alignment algorithms are known to perform poorly on non-content words, and (ii) events as well as timexes are necessarily content-bearing and hence unlikely to be realised by non-content words. This *non-content (NC) filter* is defined in terms of POS tags and affects conjunctions, prepositions and punctuation. In the context of temporal annotations, we extend the scope of the filter such that it effectively applies to all word classes that we deem unlikely to occur as part of a temporal entity. Therefore, the NC filter is actually defined stronger for events than for timexes, in that it further blocks projection of events to pronouns, whereas pronouns may be part of a timex such as *jeden Freitag 'every Friday'*. Moreover, events prohibit the projection to adverbs; this restriction is motivated by the fact that events in English are frequently translated in German as adverbials which lack an event reading (cf. head switching translations like *prefer to X* vs. German *lieber X 'rather X'*). We also devise an unknown word filter: it applies to words for which no lemma could be identified in the preprocessing stage. Projection to unknown words is prohibited unless the alignment is supported bidirectionally. The strictness concerning unknown words is due to the empirical observation that alignments which involve such words are frequently incorrect.

In order to adhere to the TimeML specification, a simple transformation ensures that articles and contracted prepositions such as *am 'on the'* are included in the extent of timexes. Another heuristics is designed to remedy alignment errors involving auxiliary and modal verbs, which are not to be annotated as events. If an event aligns to more than one word, then this filter singles out the main verb or noun and discards auxiliaries.

Topological constraints. In section 3.1, we described a conservative projection principle which rejects the transfer of annotations to non-contiguous sequences. That model sets an unnecessarily modest upper bound on recall; but giving up the contiguity requirement entirely is not sensible either, since it is indeed highly unlikely for temporal entities to be realised discontinuously in either source or target language (*noun phrase cohesion*, Yarowsky and Ngai (2001)). Based on these observations, we propose two refined models which manipulate the projected annotation span so as to ensure contiguity. One

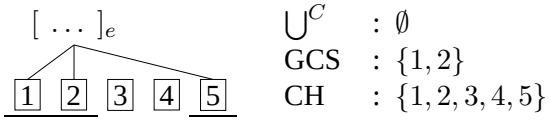
model identifies and discards *outlier alignments*, which actively violate contiguity; the other one adds *missing alignments*, which form gaps. Technically, both models establish convexity in non-convex sets. Hence, we first have to come up with a backbone model which is less restrictive than the baseline, so that the convexation models will have a basis to operate on. A possible backbone model $a_{t,0}$ is provided in (1).

$$(1) \quad a_{t,0}(e, a_s, al) = \bigcup_{w_s \in a_s(e)} \mathrm{proj}(w_s, e, a_s, al)$$

This model simply gathers all words aligned with any word covered by e in the source annotation, irrespective of contiguity in the resulting sequence of words. Discarding outlier alignments is then formalised as a reduction of $a_{t,0}$'s output to (one of) its greatest convex subset(s) (GCS). Let us call this model $a_{t,\mathrm{GCS}}$. In terms of a linear sequence of words, $a_{t,\mathrm{GCS}}$ chooses the longest contiguous subsequence. The GCS-model thus serves a filtering purpose similar to the NC filter. However, whereas the latter discards single alignment links on linguistic grounds, the former is motivated by topological properties of the alignment as a whole.

The second model, which fills gaps in the word alignment, constructs the *convex hull* of $a_{t,0}$ (cf. Padó and Lapata (2005)). We will refer to this model as $a_{t,\mathrm{CH}}$. The example in (2) illustrates both models.

(2)



$\bigcup^C$: $\emptyset$
GCS : $\{1, 2\}$
CH : $\{1, 2, 3, 4, 5\}$

Here, entity e aligns to the non-contiguous token sequence $[1, 2, 5]$, or equivalently, the non-convex set $\{1, 2, 5\}(= a_{t,0}(e))$. The conservative baseline $a_{t,\mathrm{C}}$ rejects the projection altogether, whereas $a_{t,\mathrm{GCS}}$ projects to the tokens 1 and 2. The additional padding introduced by the convex hull ($a_{t,\mathrm{CH}}$) further extends the projected extent to $\{1, 2, 3, 4, 5\}$.

Alignment selection. Although bi-alignments are known to exhibit high precision (Koehn et al., 2003), in the face of sparse annotations we use unidirectional alignments as a fallback, as has been proposed in the context of phrase-based machine translation (Koehn et al., 2003; Tillmann, 2003). Furthermore, we follow Hwa et al. (2005) in imposing a limit on the maximum number of words that a single word may align to.

4 Experiments

Our evaluation setup consists of experiments conducted on the English-German portion of the Europarl corpus (Koehn, 2005); specifically, we work with the preprocessed and word-aligned version used in Padó and Lapata (2006): the source-target and target-source word alignments were automatically established by GIZA++ (Och and Ney, 2003), and their intersection achieves a precision of 98.6% and a recall of 52.9% (Padó, 2007). The preprocessing consisted of automatic POS tagging and lemmatisation.

To assess the quality of the TimeML projections, we put aside and manually annotated a development set of 101 and a test set of 236 bi-sentences.[4] All remaining data (approx. 960K bi-sentences) was used for training (section 5). We report the weighted macro average over all possible subclasses of timexes/events, and consider only exact matches. The TARSQI annotations exhibit an F_1-measure of 80.56% (timex), 84.64% (events), and 43.32% (slinks) when evaluated against the English gold standard.

In order to assess the usefulness of the linguistic and topological parameters presented in section 3.2, we determined the best performing combination of parameters on the development set. Not surprisingly, event and timex models benefit from the various heuristics to different degrees. While the projection of events can benefit from the NC filter, the projection of timexes is rather hampered by it. Instead, it exploits the flexibility of the GCS convexation model together with a conservative limit of 2 on per-word alignments. In the underlying data sample of 101 sentences, the English-to-German alignment direction appears to be most accurate for timexes. Table 2 shows the results of evaluating the optimised models on the test set, along with the baseline from section 3.1 and a "full" model which activates all

[4] The unconventional balance of test and development data is due to the fact that a large portion of the annotated data became available only after the parameter estimation phase.

model	events			slinks			time expressions		
	prec	recall	F	prec	recall	F	prec	recall	F
timex-optimised	48.53	33.73	39.80	30.09	10.71	15.80	**71.01**	**52.76**	**60.54**
event-optimised	**50.94**	**44.23**	**47.34**	30.96	14.29	19.55	56.55	42.52	48.54
combined	**50.98**	**44.36**	**47.44**	**30.96**	**14.29**	**19.55**	**71.75**	**52.76**	**60.80**
baseline	52.26	33.46	40.80	26.98	10.71	15.34	49.53	37.80	42.87
full	51.10	40.42	45.14	29.95	13.57	18.68	73.74	54.33	62.56

Table 2: Performance of projection models over test data.

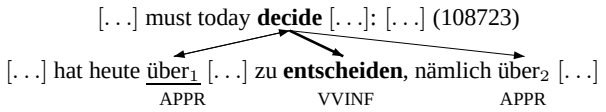


Figure 5: Amending alignment errors.

data	event		timex	
	prec	recall	prec	recall
all	53.15	45.14	73.74	53.54
best 75%	54.81	47.06	74.61	62.82

Table 3: Correlation between alignment probability and projection quality.

heuristics. The results confirm our initial assumption that linguistic and topological knowledge does indeed improve the quality of the projected annotations. The model which combines the optimal settings for timexes and events outperforms the uninformed baseline by 17.93% (timexes) and 6.64% (events) F_1-measure. However, exploration of the model space on the basis of the (larger and thus presumably more representative) test set shows that the optimised models do not generalise well. The *test set*-optimised model activates all linguistic heuristics, and employs $a_{t,\mathrm{CH}}$ convexation. For events, projection considers bi-alignments with a fallback to unidirectional alignments, preferably from English to German; timex projection considers all alignment links. This test set-optimised model, which we will use to project the training instances for the maximum entropy classifier, achieves an F_1-measure of 48.82% (53.15% precision) for events and 62.04% (73.74% precision) for timexes.[5]

With these settings, our projection model is capable of repairing alignment errors, as shown in Fig. 5, where the automatic word alignments are represented as arrows. The conservative baseline considering only bidirectional alignments discards all alignments but the (incorrect) one to *über*$_1$. The optimised model, on the other hand, does not exclude any alignments in the first place; the faulty alignments to *über*$_1$ and *über*$_2$ are discarded on linguistic grounds by the NC filter, and only the correct alignment to *entscheiden* remains for projection.

5 Robust Induction

The projected annotations, although noisy, can be exploited to train a temporal labeller for German. As Yarowsky and Ngai (2001) demonstrate for POS tagging, aggressive filtering techniques applied to vast amounts of (potentially noisy) training data are capable of distilling relatively high-quality data sets, which may then serve as input to machine learning algorithms. Yarowsky and Ngai (2001) use the Model-3 alignment score as an indicator for the quality of (i) the alignment, and therefore (ii) the projection. In the present study, discarding 25% of the sentences based on this criterion leads to gains in both recall and precision (Table 3). In accordance with the TimeML definition, we further restrict training instances on the basis of POS tags by basically re-applying the NC filter (section 3.2). But even so, the proportion of positive and negative instances remains heavily skewed—an issue which we will address below by formulating a 2-phase classi-

[5] The model actually includes an additional strategy to adjust event and timex class labels on the basis of designated FrameNet frames; the reader is referred to Spreyer (2007), ch. 4.5 for details.

model	prec	recall event	F	F slink
1-step	83.48	32.58	46.87	17.01
1-step unk	83.88	32.19	46.53	16.87
2-step	**83.95**	**34.44**	**48.84**	**19.06**
2-step unk	84.21	34.30	48.75	19.06
		timex		
1-step	87.77	49.11	62.98	
1-step unk	87.22	49.55	63.20	
2-step	**89.52**	**51.79**	**65.62**	
2-step unk	88.68	50.89	64.67	

Table 4: Classifier performance over test data.

fication task.

The remaining instances[6] are converted to feature vectors encoding standard lexical and grammatical features such as (lower case) lemma, POS, governing prepositions, verbal dependents, etc.[7] For slink instances, we further encode the syntactic subordination path (if any) between the two events.

We trained 4 classifiers,[8] with and without smoothing with artificial unknowns (Collins, 2003), and as a 1-step versus a 2-step decision in which instances are first discriminated by a binary classifier, so that only positive instances are passed on to be classified for a subclass. The performance of the various classifiers is given in Table 4. Although the overall F_1-measure does not notably differ from that achieved by direct projection, we observe a drastic gain in precision, albeit at the cost of recall. With almost 84% and 90% precision, this is an ideal starting point for a bootstrapping procedure.

6 Discussion and Future Work

Clearly, the—essentially unsupervised—projection framework presented here does not produce state-of-the-art annotations. But it does provide an inexpensive and largely language-independent basis (a) for manual correction, and (b) for bootstrapping algorithms. In the future, we will investigate how weakly supervised machine learning techniques like co-training (Blum and Mitchell, 1998) could further enhance projection, e.g. taking into account a third language in a triangulation setting (Kay, 1997).

[6] Note that slink instances are constructed for event *pairs*, as opposed to event and timex instances, which are constructed for individual words.

[7] The grammatical features have been extracted from analyses of the German ParGram LFG grammar (Rohrer and Forst, 2006).

[8] We used the `opennlp.maxent` package, `http://maxent.sourceforge.net/`.

Acknowledgements

We would like to thank Sebastian Padó for providing us with the aligned Europarl data, Inderjeet Mani and Marc Verhagen for access to the TARSQI tools, and James Pustejovsky for clarification of TimeML issues. We would also like to thank the three anonymous reviewers for helpful comments.

References

Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the 1998 Conference on Computational Learning Theory*, pages 92–100, July.

Michael Collins. 2003. Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29(4):589–637, December.

Bonnie J. Dorr. 1994. Machine Translation Divergences: A Formal Description and Proposed Solution. *Computational Linguistics*, 20(4):597–635.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson, 2005. *TIDES 2005 Standard for the Annotation of Temporal Expressions*, September.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating Translational Correspondence using Annotation Projection. In *Proceedings of ACL-2002*, Philadelphia, PA.

R. Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping Parsers via Syntactic Projection across Parallel Texts. *Natural Language Engineering*, 11(3):311–325.

Martin Kay. 1997. The Proper Place of Men and Machines in Language Translation. *Machine Translation*, 12(1-2):3–23.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT/NAACL 2003*, pages 127–133.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit 2005*.

Inderjeet Mani and George Wilson. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pages 69–76, Hong Kong.

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. 2003. Inferring Temporal Ordering of Events in News. In *Proceedings of the Human Language Technology Conference (HLT-NAACL-2003)*. Short paper.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine Learning of Temporal Relations. In *Proceedings of ACL/COLING 2006*, pages 753–760, Sydney, Australia.

Franz J. Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Sebastian Padó and Mirella Lapata. 2005. Cross-lingual projection of role-semantic information. In *Proceedings of HLT/EMNLP 2005*, Vancouver, BC.

Sebastian Padó and Mirella. Lapata. 2006. Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of ACL-COLING 2006*, Sydney, Australia.

Sebastian Padó. 2007. *Cross-Lingual Annotation Projection Models for Role-Semantic Information*. Ph.D. thesis, Saarland University, Saarbrücken, Germany.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of the Fifth International Workshop on Computational Semantics*.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003b. The TimeBank Corpus. In *Proceedings of Corpus Linguistics*, pages 647–656.

Christian Rohrer and Martin Forst. 2006. Improving coverage and parsing quality of a large-scale LFG for German. In *Proceedings of LREC 2006*, pages 2206–2211, Genoa, Italy, May.

Estela Saquete, Patricio Martínez-Barco, Rafael Muñoz, Matteo Negri, Manuela Speranza, and Rachele Sprugnoli. 2006. Multilingual Extension of a Temporal Expression Normalizer using Annotated Corpora. In *Proceedings of the EACL 2006 Workshop on Cross-Language Knowledge Induction*, Trento, Italy, April.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005a. Evita: A Robust Event Recognizer For QA Systems. In *Proceedings of HLT/EMNLP 2005*, pages 700–707.

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky, 2005b. *TimeML Annotation Guidelines Version 1.2.1*, October.

Roser Saurí, Marc Verhagen, and James Pustejovsky. 2006. SlinkET: A Partial Modal Parser for Events. In *Proceedings of LREC-2006*, Genova, Italy, May. To appear.

Amanda Schiffrin and Harry Bunt. 2006. Defining a preliminary set of interoperable semantic descriptors. Technical Report D4.2, INRIA-Loria, Nancy, France, August.

Andrea Setzer. 2001. *Temporal Information in Newswire Articles: an Annotation Scheme and Corpus Study*. Ph.D. thesis, University of Sheffield, Sheffield, UK.

Kathrin Spreyer. 2007. Projecting Temporal Annotations Across Languages. Diploma thesis, Saarland University, Saarbrücken, Germany.

Christoph Tillmann. 2003. A Projection Extension Algorithm for Statistical Machine Translation. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP-2003)*, pages 1–8.

Zeno Vendler, 1967. *Linguistics in Philosophy*, chapter Verbs and Times, pages 97–121. Cornell University Press, Ithaca, NY.

Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Jessica Littman, and James Pustejovsky. 2005. Automating Temporal Annotation with TARSQI. In *Proceedings of the ACL-2005*.

David Yarowsky and Grace Ngai. 2001. Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In *Proceedings of NAACL-2001*, pages 200–207.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*.

Acquiring Event Relation Knowledge by Learning Cooccurrence Patterns and Fertilizing Cooccurrence Samples with Verbal Nouns

Shuya Abe **Kentaro Inui** **Yuji Matsumoto**
Graduate School of Information Science,
Nara Institute of Science and Technology
{shuya-a,inui,matsu}@is.naist.jp

Abstract

Aiming at acquiring semantic relations between events from a large corpus, this paper proposes several extensions to a state-of-the-art method originally designed for entity relation extraction, reporting on the present results of our experiments on a Japanese Web corpus. The results show that (a) there are indeed specific cooccurrence patterns useful for event relation acquisition, (b) the use of cooccurrence samples involving verbal nouns has positive impacts on both recall and precision, and (c) over five thousand relation instances are acquired from a 500M-sentence Web corpus with a precision of about 66% for *action-effect* relations.

1 Introduction

The growing interest in practical NLP applications such as question answering, information extraction and multi-document summarization places increasing demands on the processing of relations between textual fragments such as entailment and causal relations. Such applications often need to rely on a large amount of lexical semantic knowledge. For example, a causal (and entailment) relation holds between the verb phrases *wash something* and *something is clean*, which reflects the commonsense notion that if someone has washed something, this object is clean as a result of the washing event. A crucial issue is how to obtain and maintain a potentially huge collection of such event relations instances.

Motivated by this background, several research groups have reported their experiments on automatic acquisition of causal, temporal and entailment relations between event mentions (typically verbs or verb phrases) (Lin and Pantel, 2001; Inui et al., 2003; Chklovski and Pantel, 2005; Torisawa, 2006; Pekar, 2006; Zanzotto et al., 2006, etc.). The common idea behind them is to use a small number of manually selected generic lexico-syntactic cooccurrence patterns (LSPs or simply patterns). *to Verb-X and then Verb-Y*, for example, is used to obtain temporal relations such as *marry* and *divorce* (Chklovski and Pantel, 2005). The use of such generic patterns, however, tends to be high recall but low precision, which requires an additional component for pruning extracted relations. This issue has been addressed in basically two approaches, either by devising heuristic statistical scores (Chklovski and Pantel, 2005; Torisawa, 2006; Zanzotto et al., 2006) or training classifiers for disambiguation with heavy supervision (Inui et al., 2003).

This paper explores a third way for enhancing present LSP-based methods for event relation acquisition. The basic idea is inspired by the following recent findings in relation extraction (Ravichandran and Hovy, 2002; Pantel and Pennacchiotti, 2006, etc.), which aims at extracting semantic relations between *entities* (as opposed to *events*) from texts. (a) The use of generic patterns tends to be high recall but low precision, which requires an additional component for pruning. (b) On the other hand, there are specific patterns that are highly reliable but they are much less frequent than generic patterns and each makes only a small contribution to recall. (c) Combining a few generic patters with a much larger collection of reliable specific patterns boosts both pre-

cision and recall. Such specific patterns can be acquired from a very large corpus with seeds.

Given these insights, an intriguing question is whether the same story applies to event relation acquisition as well or not. In this paper, we explore this issue through the following steps. First, while previous methods use only verb-verb cooccurrences, we use cooccurrences between verbal nouns and verbs such as *cannot ⟨find out (something)⟩ due to the lack of ⟨investigation⟩* as well as verb-verb cooccurrences. This extension dramatically enlarge the pool of potential candidate LSPs (Section 4.1). Second, we extend Pantel and Pennacchiotti (2006)'s Espresso algorithm, which induces specific reliable LSPs in a bootstrapping manner for entity relation extraction, so that the extended algorithm can apply to event relations (Sections 4.2 to 4.4). Third, we report on the present results of our empirical experiments, where the extended algorithm is applied to a Japanese 500M-sentence Web corpus to acquire two types of event relations, *action-effect* and *action-means* relations (Section 5)

2 Related work

Perhaps a simplest way of using LSPs for event relation acquisition can be seen in the method Chklovski and Pantel (2005) employ to develop VerbOcean. Their method uses a small number of manually selected generic LSPs such as *to Verb-X and then Verb-Y* to obtain six types of semantic relations including *strength* (e.g. *taint – poison*) and *happens-before* (e.g. *marry – divorce*) and obtain about 29,000 verb pairs with 65.5% precision.

One way for pruning extracted relations is to incorporate a classifier trained with supervision. Inui et al. (2003), for example, use a Japanese generic causal connective marker *tame* (because) and a supervised classifier learner to separately obtain four types of causal relations: *cause*, *precondition*, *effect* and *means*.

Torisawa (2006), on the other hand, acquires entailment relations by combining the verb pairs extracted with a highly generic connective pattern *Verb-X and Verb-Y* together with the cooccurrence statistics between verbs and their arguments. While the results Torisawa reports look promising, it is not clear yet if the method applies to other types of relations because it relies on relation-specific heuristics.

Another direction from (Chklovski and Pantel, 2005) is in the use of LSPs involving nominalized verbs. Zanzotto et al. (2006) obtain, for example, an entailment relation *X wins → X plays* from such a pattern as *player wins*. However, their way of using nominalized verbs is highly limited compared with our way of using verbal nouns.

3 Espresso

This section overviews Pantel and Pennacchiotti (2006)'s Espresso algorithm. Espresso takes as input a small number of seed instances of a given target relation and iteratively learns cooccurrence patterns and relation instances in a bootstrapping manner.

Ranking cooccurrence patterns For each given relation instance $\{x, y\}$, Espresso retrieves the sentences including both x and y from a corpus and extracts from them cooccurrence samples. For example, given an instance of the *is-a* relation such as ⟨*Italy,country*⟩, Espresso may find cooccurrence samples such as *countries such as Italy* and extract such a pattern as *Y such as X*. Espresso defines the reliability $r_\pi(p)$ of pattern p as the average strength of its association with each relation instance i in the current instance set I, where each instance i is weighted by its reliability $r_\iota(i)$:

$$r_\pi(p) = \frac{1}{|I|} \sum_{i \in I} \frac{pmi(i,p)}{max_{pmi}} \times r_\iota(i) \qquad (1)$$

where $pmi(i,p)$ is the pointwise mutual information between i and p, and max_{pmi} is the maximum PMI between all patterns and all instances.

Ranking relation instances Intuitively, a reliable relation instance is one that is highly associated with multiple reliable patterns. Hence, analogously to the above pattern reliability measure, Espresso defines the reliability $r_\iota(i)$ of instance i as:

$$r_\iota(i) = \frac{1}{|P|} \sum_{p \in P} \frac{pmi(i,p)}{max_{pmi}} \times r_\pi(p) \qquad (2)$$

where $r_\pi(p)$ is the reliability of pattern p, defined above in (1), and max_{pmi} is as before. $r_\iota(i)$ and $r_\pi(p)$ are recursively defined, where $r_\iota(i) = 1$ for each manually supplied seed instance i[1].

[1]For our extension, $r_\iota(i) = -1$ for each manually supplied negative instance.

4 Event relation acquisition

Our primary concerns are whether there are indeed specific cooccurrence patterns useful for acquiring event relations and whether such patterns can be found in a bootstrapping manner analogous to Espresso. To address these issues, we make several extensions to Espresso, which is originally designed for entity relations (not scoping event relations).

4.1 Cooccurences with verbal nouns

Most previous methods for event relation acquisition rely on verb-verb cooccurrences because verbs (or verb phrases) are the most typical device for referring to events. However, languages have another large class of words for event reference, namely verbal nouns or nominalized forms of verbs. In Japanese, for example, verbal nouns such as *kenkyu* (research) constitute the largest morphological category used for event reference.

Japanese verbal nouns have dual statuses, as verbs and nouns. When occurring with the verb *suru* (do-PRES), verbal nouns function as a verb as in (1a). On the other hand, when accompanied by case markers such as *ga* (NOMINATIVE) and *o* (ACCUSATIVE), they function as a noun as in (1b). Finally, but even more importantly, when accompanied by a large variety of suffixes, verbal nouns constitute compound nouns highly productively as in (1c).

(1) a. *Ken-ga* *gengo-o* *kenkyu-suru*
Ken-NOM language-ACC research-PRES
Ken researches on language.

b. *Ken-ga* *gengo-no* *kenkyu-o* *yame-ta*
Ken-NOM language-on research-ACC quit-PAST
Ken quitted research on language.

c. *-sha* (person):
e.g. *kenkyu-sha* (researcher)
-shitsu (place):
e.g. *kenkyu-shitsu* (laboratory)
-go (after):
e.g. *kenkyu-go* (after research)

These characteristics of verbal nouns can be made use of to substantially increase both cooccurrence instances and candidate cooccurrence patterns (see Section 5.1 for statistics). For example, the verbal noun *kenkyu* (research) often cooccurs with the verb *jikken* (experiment) in the pattern of (2a). From those cooccurrences, one may learn that *jikken-suru* (to experiment) is an action that is often taken as a part of *kenkyu-suru* (to research). In such a case, we may consider a pattern as shown in (2b) useful for acquiring *part-of* relations between actions.

(2) a. *kenkyu-shitsu-de* *jikken-suru*
research-place-in experiment-VERB
conduct experiments in the laboratory

b. *(Act-X)-shitsu-de* *(Act-Y)-suru*
(Act-X)-place-in (Act-X)-VERB
(Act-Y) is often done in doing (Act-X)

When functioning as a noun, verbal nouns are potentially ambiguous between the event reading and the entity/object reading. For example, the verbal noun *denwa* (phone) in the context *denwa-de* (phone-by) may refer to either a phone-call event or a physical phone. While, ideally, such eventhood ambiguities should be resolved before collecting cooccurrence samples with verbal nouns, we simply use all the occurrences of verbal nouns in collecting cooccurrences in our experiments. It is an interesting issue for future work whether eventhood determination would have a strong impact on the performance of event relation extraction.

4.2 Selection of arguments

One major step from the extraction of entity relations to the extraction of event relations is how to address the issue of *generalization*. In entity relation extraction, relations are typically assumed to hold between chunks like named entities or simply between one-word terms, where the issue of determining the appropriate level of the generality of extracted relations has not been salient. In event relation extraction, on the other hand, this issue immediately arises. For example, the cooccurrence sample in (3) suggests the *action-effect* relation between *niku-o yaku* (grill the meat) and *(niku-ni) kogeme-ga tsuku* ((the meat) gets brown)[2].

(3) *(kogeme-ga* *tsuku) -kurai* *niku-o* *yaku*
a burn-NOM get -so that meat-ACC grill
grill the meat so that it gets brown
(grill the meat to a deep brown)

In this relation, the argument *niku* (meat) of the verb *yaku* (grill) can be dropped and generalized

[2]The parenthesis in the first row of (3) indicates a subordinate clause.

to *something to grill*; namely the *action-effect* relation still holds between *X-o yaku* (grill X) and *X-ni kogeme-ga tsuku* (X gets brown). On the other hand, however, the argument *kogeme* (a burn) of the verb *tsuku* (get) cannot be dropped; otherwise, the relation would no longer hold.

One straightforward way to address this problem is to expand each cooccurrence sample to those corresponding to different degrees of generalization and feed them to the relation extraction model so that its scoring function can select appropriate event pairs from expanded samples. For example, cooccurrence sample (3) is expanded to those as in (4):

(4) a. *(kogeme-ga tsuku) -kurai niku-o yaku*
a burn-NOM get -so that meat-ACC grill

b. *(tsuku) -kurai niku-o yaku*
get -so that meat-ACC grill

c. *(kogeme-ga tsuku) -kurai yaku*
a burn-NOM get -so that grill

d. *(tsuku) -kurai yaku*
get -so that grill

In practice, in our experiments (Section 5), we restrict the number of arguments for each event up to one to avoid the explosion of the types of infrequent candidate relation instances.

4.3 Volitionality of events

Inui et al. (2003) discuss how causal relations between events should be typologized for the purpose of semantic inference and classify causal relations basically into four types — Effect, Means, Precondition and Cause relations — based primarily on the volitionality of involved events. For example, Effect relations hold between volitional actions and their resultative non-volitional states/happenings/experiences, while Cause relations hold between only non-volitional states/happenings/experiences.

Following this typology, we are concerned with the volitionality of each event mention. For our experiments, we manually built a lexicon of over 12,000 verbs (including verbal nouns) with volitionality labels, obtaining 8,968 volitional verbs, 3,597 non-volitional and 547 ambiguous. Volitional verbs include *taberu* (eat) and *kenkyu-suru* (research), while non-volitional verbs include *atatamaru* (get warm), *kowareru* (to break-*vi*) and *kanashimu* (be sad). We discarded the ambiguous verbs in the experiments.

4.4 Dependency-based cooccurrence patterns

The original Espresso encodes patterns simply as a word sequence because entity mentions in the relations it scopes tend to cooccur locally in a single phrase or clause. In event relation extraction, however, cooccurrence patterns of event mentions in the relations we consider (causal relations, temporal relations, etc.) can be captured better as a path on a syntactic dependency tree because (i) such mention pairs tend to cooccur in a longer dependency path and (ii) as discussed in Section 4.2, we want to exclude the arguments of event mentions from cooccurrence patterns, which would be difficult with word sequence-based representations of patterns.

A Japanese sentence can be analyzed as a sequence of base phrase (BP) chunks called *bunsetsu* chunks, each which typically consists of one content (multi-)word followed by functional words. We assume each sentence of our corpus is given a dependency parse tree over its BP chunks. Let us call a BP chunk containing a verb or verbal noun an *event chunk*. We create a cooccurrence sample from any pair of event chunks that cooccur if either (a) one event chunk depends directly on the other, or (b) one event chunk depends indirectly on the other via one intermediate chunk. Additionally, we apply the Japanese functional expressions dictionary (Matsuyoshi et al., 2006) to a cooccurrence pattern for generalization.

In (5), for example, the two event chunks, *taishoku-go-ni* (after retirement) and *hajimeru* (begin), meet the condition (b) above and the dependency path designated by bold font is identified as a candidate cooccurrence pattern. The argument *PC-o* of the verb *hajimeru* is excluded from the path.

(5) *(taishoku-**go-no** **tanoshimi)-ni** PC-o hajimeru*
retirement-after as a hobby PC-ACC begin
begin a PC as a hobby after retirement

5 Experiments

5.1 Settings

For an empirical evaluation, we used a sample of approximately 500M sentences taken from the

Table 1: Examples of acuired cooccurrence patterns and relatio instances for the action-effect relation

freq	cooccurrence patterns	relation instances
94477	⟨*verb;action*⟩***temo***⟨*verb;effect*⟩**nai** (to do ⟨action⟩ though ⟨effect⟩ dose not happen)	*sagasu::mitsukaru* (search::be found), *asaru::mitsukaru* (hunt::be found), *purei-suru::kuria-suru* (play::finish)
6250	⟨*verb;action*⟩***takeredomo***⟨*verb;effect*⟩**nai** (to do ⟨action⟩ though ⟨effect⟩ dose not happen)	*shashin-wo-toru::toreru* (shot photograph::be shot), *meiru-wo-okuru::henji-ga-kaeru* (send a mail::get an answer)
1851	⟨*noun;action*⟩***wo-shitemo***⟨*verb;effect*⟩**nai** (to do ⟨action⟩ though ⟨effect⟩ dose not happen)	*setsumei-suru::nattoku-suru* (explain::agree), *siai-suru::katsu* (play::win), *siai-suru::makeru* (play::lose)
1329	⟨*verb;action*⟩***yasukute***⟨*adjective;effect*⟩ (to simply do ⟨action⟩ and ⟨effect⟩)	*utau::kimochiyoi* (sing::feel good), *hashiru::kimochiyoi* (run::feel good)
4429	⟨*noun;action*⟩***wo-kiite***⟨*verb;effect*⟩ (to hear ⟨action⟩ so that ⟨effect⟩)	*setsumei-suru::nattoku-suru* (explain::agree), *setsumei-suru::rikai-dekiru* (explain::can understand)

Web corpus collected by Kawahara and Kurohashi (2006). The sentences were dependency-parsed with Cabocha (Kudo and Matsumoto, 2002), and cooccurrence samples of event mentions were extracted. Event mentions with patterns whose frequency was less than 20 were discarded in order to reduce computational costs. As a result, we obtained 34M cooccurrence tokens with 11M types. Note that among those cooccurrence samples 15M tokens (44%) with 4.8M types (43%) are those with verbal nouns, suggesting the potential impacts of using verbal nouns.

In our experiments, we considered two of Inui et al. (2003)'s four types of causal relations: *action-effect* relations (Effect in Inui et al.'s terminology) and *action-means* relations (Means). An *action-effect* relation holds between events x and y if and only if non-volitional event y is likely to happen as either a direct or indirect effect of volitional action x. For example, the action *X-ga undou-suru* (X exercises) and the event *X-ga ase-o kaku* (X sweats) are considered to be in this type of relation. A *action-means* relation holds between events x and y if and only if volitional action y is likely to be done as a part/means of volitional action x. For example, if case a event-pair is *X-ga hashiru* (X runs) is considered as a typical action that is often done as a part of the action *X-ga undou-suru* (X exercises).

Note that in these experiments we do not differentiate between relations with the same subject and those with a different subject. However we plan to conduct further experiments in the future that make use of this distinction.

In addition, we have collected *action-effect* relation instances for a baseline measure. The baseline consists of instances that cooccur with eleven patterns that indicate *action-effect* relation. The difference between the extended Espresso and baseline is caused by the low number and constant scores of patterns.

5.2 Results

We ran the extended Espresso algorithm starting with 971 positive and 1069 negative seed relation instances for *action-effect* relation and 860 positive and 74 negative seed relations for *action-means* relation. As a result, we obtained 34,993 cooccurrence patterns with 173,806 relation instances for the *action-effect* relation and 23,281 coocurrence relations with 237,476 relation instances for the *action-means* relation after 20 iterations of pattern ranking/selection and instance ranking/selection. The threshold parameters for selecting patterns and instances were decided in a preliminary trial. Some of the acquired patterns and instances for the *action-effect* relation are shown in Table 1.

5.2.1 Precision

To estimate precision, 100 relation instances were randomly sampled from each of four sections of the ranks of the acquired instances for each of the two relations (1–500, 501–1500, 1501–3500 and 3500–7500), and the correctness of each sampled instance was judged by two graduate students (i.e. 800 relation instances in total were judged).

Note that in these experiments we asked the assessors to both (a) the degree of the likeliness that the effect/means takes place and (b) which arguments are shared between the two events. For example, while *nomu* (drink) does not necessarily result in

futsukayoi-ni naru (have a hangover), the assessors judged this pair correct because one can at least say that the latter *sometimes* happens *as a result of* the former. For criterion (b), as shown in Table 1, the relation instances judged correct include both the *X-ga* VP_1*::X-ga* VP_2 type (i.e. two subjects are shared) and the *X-o* VP_1*::X-ga* VP_2 type (the object of the former and the subject of the latter are shared). The issue of how to control patterns of argument sharing is left for future work.

The precision for the assessed samples are shown in Figures 1 to 3. "2 judges" means that an instance is acceptable to both judges. "1 judges" means that it is an acceptable instance to at least one of the two judges. "strict" indicates correct instance relations while "lenient"[3] indicates correct instance relations – when a judge appends the right cases.

As a result of this strictness in judgement, the inter-assessor agreement turned out to be poor. The kappa statistics was 0.53 for the *action-effect* relations, 0.49 for the *action-effect* relations (=baseline) and 0.55 for *action-means* relations.

The figures show that both types of relations were acquired with reasonable precision not only for the higher-ranked instances but also for lower-ranked instances. It may seem strange that the precision of the lower-ranked *action-means* instances is sometimes even better than the higher-ranked ones, which may mean that the scoring function given in Section 3 did not work properly. While further investigation is clearly needed, it should also be noted that higher-ranked instances tended to be more specific than lower-ranked ones.

5.2.2 Effects of seed number

We reran the extended Espresso algorithm for the *action-effect* relation, starting with 500 positive and 500 negative seed relation instances. The precision is shown in Figure 4[4]. This precision is fairly lower than that of *action-effect* relations with all seed instances. Additionally, the number of seed instances affects the precision of both higher-ranked and lower-ranked instances. This result indicates that while the proposed algorithm is designed to work with a small seed set, in reality its performance severely depends on the number of seeds.

5.2.3 Effects of using verbal nouns

We also examine the effect of using verbal nouns. Of the 500 highest scored patterns for the *action-effect* relation, 128 patterns include verbal noun slots, and for *action-means*, 495 patterns. Hence, the presence of verbal nouns greatly effects some acquired instances. Additionally, to see the influence of frequency, of the 500 high frequent patterns selected from the 2000 highest scored patterns for *action-effect* relation, 177 include verbal noun slots, and for *action-means*, 407 patterns. This result provides further evidence that the inclusion of verbal nouns has a positive effect in this task.

5.2.4 Argument selection

According to our further investigation on argument selection, 49 instances (12%) of the correct *action-effect* relation instances that are judged correct have a specific argument in at least one event, and all of them would be judged incorrect (i.e. overgeneralized) if they did not have those arguments (Recall the example of *kogeme-ga tsuku* (get brown) in Section 4.2). This figure indicates that our method for argument selection works to a reasonable degree.

However, clearly there is still much room for improvement. According to our investigation, up to 26% of the instances that are judged incorrect could be saved if appropriate arguments were selected. For example, *X-ga taberu* (X eats) and *X-ga shinu* (X dies) would constitute an *action-effect* relation if the former event took such an argument as *dokukinoko-o* (toadstool-ACC). The overall precision could be boosted if an effective method for argument selection method were devised.

6 Conclusion and future work

In this paper, we have addressed the issue of how to learn lexico-syntactic patterns useful for acquiring event relation knowledge from a large corpus, and proposed several extensions to a state-of-the-art method originally designed for entity relation extraction, reporting on the present results of our empirical evaluation. The results have shown that (a) there are indeed specific cooccurrence patterns useful for event relation acquisition, (b) the use of cooccurrence samples involving verbal nouns has pos-

[3] If an instance is judged as "strict" by one assessor and "lenient" by the other, then the instance is assessed as "lenient".

[4] It was only judged by one assessor.

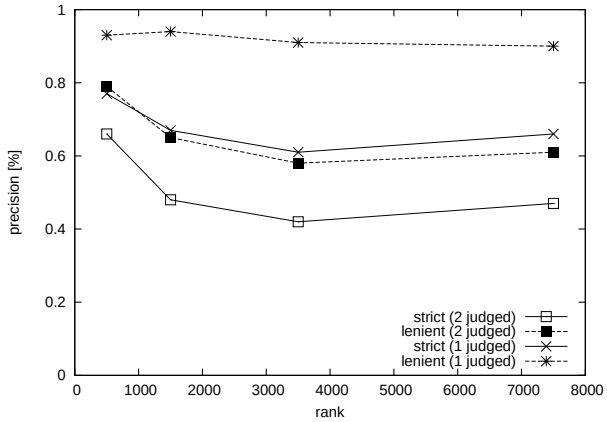


Figure 1: *action-effect*

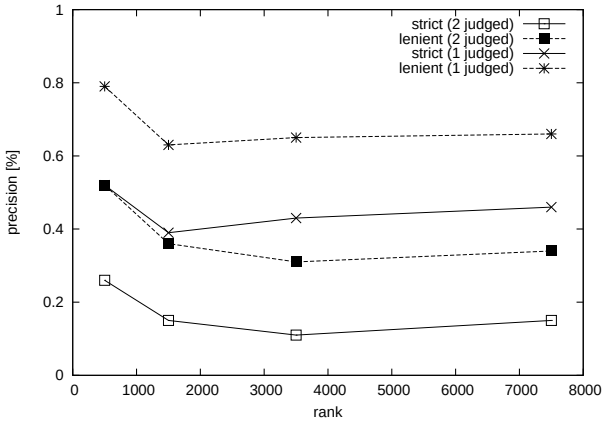


Figure 2: *action-means*

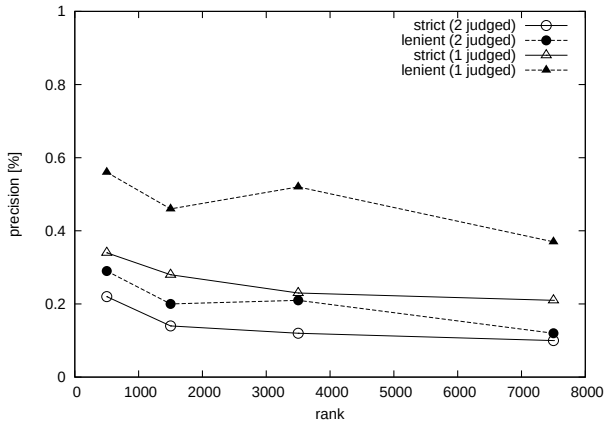


Figure 3: *action-effect* (baseline)

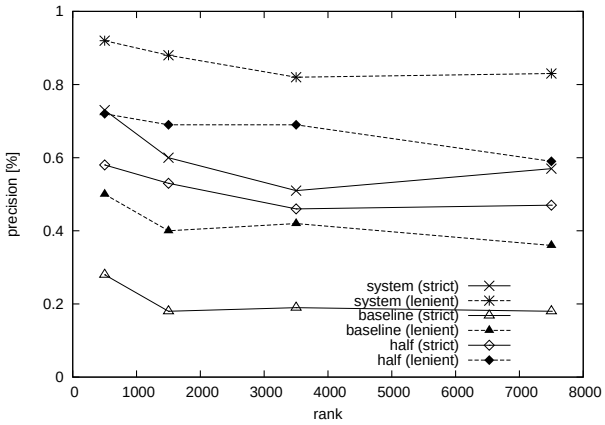


Figure 4: *action-effect* (half seed)

itive impacts on both recall and precision, and (c) over five thousand relation instances are acquired from the 500M-sentence Web corpus with a precision of about 66% for *action-effect* relations.

Clearly, there is still much room for exploration and improvement. First of all, more comprehensive evaluations need to be done. For example, the acquired relations should be evaluated in terms of recall and usefulness. A deep error analysis is also needed. Second, the experiments have revealed that one major problem to challenge is how to optimize argument selection. We are seeking a way to incorporate a probabilistic model of predicate-argument cooccurrences into the ranking function for relation instances. Related to this issue, it is also crucial to devise a method for controlling argument sharing patterns. One possible approach is to employ state-of-the-art techniques for coreference and zero-anaphora resolution (Iida et al., 2006; Komachi et al., 2007, etc.) in preprocessing cooccurrence samples.

References

Timothy Chklovski and Patrick Pantel. 2005. Global path-based refinement of noisy graphs applied to verb semantics. In *Proceedings of Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 792–803.

Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2006. Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 625–632.

Takashi Inui, Kentaro Inui, and Yuji Matsumoto. 2003. What kinds and amounts of causal knowledge can be acquired from text by using connective markers as clues? In *Proceedings of the 6th International Conference on Discovery Science*, pages 180–193. An extended version: Takashi Inui, Kentaro Inui, and Yuji Matsumoto (2005). Acquiring causal knowledge from text using the connective marker tame. *ACM Transactions on Asian Language Information Processing (TALIP), 4(4):435–474.*

Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 176–183.

Mamoru Komachi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. 2007. Learning based argument structure analysis of event-nouns in japanese. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 120–128.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69.

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2001*, pages 323–328.

Suguru Matsuyoshi, Satoshi Sato, and Takehito Utsuro. 2006. Compilation of a dictionary of japanese functional expressions with hierarchical organization. In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*, pages 395–402.

Patric Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 113–120.

Viktor Pekar. 2006. Acquisition of verb entailment from text. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 49–56.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 21st International Conference on Computational Linguistics and 40th Annual Meeting of the Association for Computational Linguistics*, pages 41–47.

Kentaro Torisawa. 2006. Acquiring inference rules with temporal constraints by using japanese coordinated sentences and noun-verb co-occurrences. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 57–64.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Maria Teresa Pazienza. 2006. Discovering asymmetric entailment relations between verbs using selectional preferences. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 849–856.

Refinements in BTG-based Statistical Machine Translation

Deyi Xiong, Min Zhang, Aiti Aw
Human Language Technology
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
{dyxiong, mzhang, aaiti}@i2r.a-star.edu.sg

Haitao Mi, Qun Liu and Shouxun Lin
Key Lab of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
Beijing China, 100080
{htmi, liuqun, sxlin}@ict.ac.cn

Abstract

Bracketing Transduction Grammar (BTG) has been well studied and used in statistical machine translation (SMT) with promising results. However, there are two major issues for BTG-based SMT. First, there is no effective mechanism available for predicting orders between neighboring blocks in the original BTG. Second, the computational cost is high. In this paper, we introduce two refinements for BTG-based SMT to achieve better reordering and higher-speed decoding, which include (1) reordering heuristics to prevent incorrect swapping and reduce search space, and (2) special phrases with tags to indicate sentence beginning and ending. The two refinements are integrated into a well-established BTG-based Chinese-to-English SMT system that is trained on large-scale parallel data. Experimental results on the NIST MT-05 task show that the proposed refinements contribute significant improvement of 2% in BLEU score over the baseline system.

1 Introduction

Bracket transduction grammar was proposed by Wu (1995) and firstly employed in statistical machine translation in (Wu, 1996). Because of its good trade-off between efficiency and expressiveness, BTG restriction is widely used for reordering in SMT (Zens et al., 2004). However, BTG restriction does not provide a mechanism to predict final orders between two neighboring blocks.

To solve this problem, Xiong et al. (2006) proposed an enhanced BTG with a maximum entropy (MaxEnt) based reordering model (MEBTG). MEBTG uses boundary words of bilingual phrases as features to predict their orders. Xiong et al. (2006) reported significant performance improvement on Chinese-English translation tasks in two different domains when compared with both Pharaoh (Koehn, 2004) and the original BTG using flat reordering. However, error analysis of the translation output of Xiong et al. (2006) reveals that boundary words predict wrong swapping, especially for long phrases although the MaxEnt-based reordering model shows better performance than baseline reordering models.

Another big problem with BTG-based SMT is the high computational cost. Huang et al. (2005) reported that the time complexity of BTG decoding with m-gram language model is $O(n^{3+4(m-1)})$. If a 4-gram language model is used (common in many current SMT systems), the time complexity is as high as $O(n^{15})$. Therefore with this time complexity translating long sentences is time-consuming even with highly stringent pruning strategy.

To speed up BTG decoding, Huang et al. (2005) adapted the hook trick which changes the time complexity from $O(n^{3+4(m-1)})$ to $O(n^{3+3(m-1)})$. However, the implementation of the hook trick with pruning is quite complicated. Another method to increase decoding speed is cube pruning proposed by Chiang (2007) which reduces search space significantly.

In this paper, we propose two refinements to address the two issues, including (1) reordering heuris-

tics to prevent incorrect swapping and reduce search space using swapping window and punctuation restriction, and (2) phrases with special tags to indicate beginning and ending of sentence. Experimental results show that both refinements improve the BLEU score significantly on large-scale data.

The above refinements can be easily implemented and integrated into a baseline BTG-based SMT system. However, they are not specially designed for BTG-based SMT and can also be easily integrated into other systems with different underlying translation strategies, such as the state-of-the-art phrase-based system (Koehn et al., 2007), syntax-based systems (Chiang et al., 2005; Marcu et al., 2006; Liu et al., 2006).

The rest of the paper is organized as follows. In section 2, we review briefly the core elements of the baseline system. In section 3 we describe our proposed refinements in detail. Section 4 presents the evaluation results on Chinese-to-English translation based on these refinements as well as results obtained in the NIST MT-06 evaluation exercise. Finally, we conclude our work in section 5.

2 The Baseline System

In this paper, we use Xiong et al. (2006)'s system Bruin as our baseline system. Their system has three essential elements which are (1) a stochastic BTG, whose rules are weighted using different features in log-linear form, (2) a MaxEnt-based reordering model with features automatically learned from bilingual training data, (3) a CKY-style decoder using beam search similar to that of Wu (1996). We describe the first two components briefly below.

2.1 Model

The translation process is modeled using BTG rules which are listed as follows

$$A \rightarrow [A^1, A^2] \tag{1}$$

$$A \rightarrow \langle A^1, A^2 \rangle \tag{2}$$

$$A \rightarrow x/y \tag{3}$$

The lexical rule (3) is used to translate source phrase x into target phrase y and generate a block A. The two rules (1) and (2) are used to merge two consecutive blocks into a single larger block in a straight or inverted order.

To construct a stochastic BTG, we calculate rule probabilities using the log-linear model (Och and Ney, 2002). For the two merging rules (1) and (2), the assigned probability $Pr^m(A)$ is defined as follows

$$Pr^m(A) = \Omega^{\lambda_\Omega} \cdot \triangle_{p_{LM}(A^1,A^2)}^{\lambda_{LM}} \tag{4}$$

where Ω, the reordering score of block A^1 and A^2, is calculated using the MaxEnt-based reordering model (Xiong et al., 2006) described in the next section, λ_Ω is the weight of Ω, and $\triangle_{p_{LM}(A^1,A^2)}$ is the increment of language model score of the two blocks according to their final order, λ_{LM} is its weight.

For the lexical rule (3), it is applied with a probability $Pr^l(A)$

$$\begin{aligned} Pr^l(A) &= p(x|y)^{\lambda_1} \cdot p(y|x)^{\lambda_2} \cdot p_{lex}(x|y)^{\lambda_3} \\ &\quad \cdot p_{lex}(y|x)^{\lambda_4} \cdot exp(1)^{\lambda_5} \cdot exp(|y|)^{\lambda_6} \\ &\quad \cdot p_{LM}^{\lambda_{LM}}(y) \end{aligned} \tag{5}$$

where $p(\cdot)$ are the phrase translation probabilities in both directions, $p_{lex}(\cdot)$ are the lexical translation probabilities in both directions, $exp(1)$ and $exp(|y|)$ are the phrase penalty and word penalty, respectively and λs are weights of features. These features are commonly used in the state-of-the-art systems (Koehn et al., 2005; Chiang et al., 2005).

2.2 MaxEnt-based Reordering Model

The MaxEnt-based reordering model is defined on two consecutive blocks A^1 and A^2 together with their order $o \in \{straight, inverted\}$ according to the maximum entropy framework.

$$\Omega = p_\theta(o|A^1, A^2) = \frac{exp(\sum_i \theta_i h_i(o, A^1, A^2))}{\sum_o exp(\sum_i \theta_i h_i(o, A^1, A^2))} \tag{6}$$

where the functions $h_i \in \{0, 1\}$ are model features and θ_i are weights of the model features trained automatically (Malouf, 2002).

There are three steps to train a MaxEnt-based reordering model. First, we need to extract reordering examples from unannotated bilingual data, then generate features from these examples and finally estimate feature weights.

For extracting reordering examples, there are two points worth mentioning:

1. In the extraction of useful reordering examples, there is no length limitation over blocks compared with extracting bilingual phrases.

2. When enumerating all combinations of neighboring blocks, a good way to keep the number of reordering examples acceptable is to extract smallest blocks with the *straight* order while largest blocks with the *inverted* order .

3 Refinements

In this section we describe two refinements mentioned above in detail. First, we present fine-grained reordering heuristics using swapping window and punctuation restriction. Secondly, we integrate special bilingual phrases with sentence beginning/ending tags.

3.1 Reordering Heuristics

We conduct error analysis of the translation output of the baseline system and observe that Bruin sometimes incorrectly swaps two large neighboring blocks on the target side. This happens frequently when inverted order successfully challenges straight order by the incorrect but strong support from the language model and the MaxEnt-based reordering model. The reason is that only boundary words are used as evidences by both language model and MaxEnt-based reordering model when the decoder selects which merging rule (straight or inverted) to be used [1]. However, statistics show that boundary words are not reliable for predicting the right order between two larger neighboring blocks. Al-Onaizan and Papineni (2006) also proved that language model is insufficient to address long-distance word reordering. If a wrong inverted order is selected for two large consecutive blocks, incorrect long-distance swapping happens.

Yet another finding is that many incorrect swappings are related to punctuation marks. First, the source sequence within a pair of balanced punctuation marks (quotes and parentheses) should be kept within the punctuation after translation. However, it is not always true when reordering is involved. Sometime the punctuation marks are distorted with the enclosed words sequences being moved out. Secondly, it is found that a series of words is frequently reordered from one side of a structural mark, such as commas, semi-colons and colons, to the other side of the mark for long sentences containing such marks. Generally speaking, on Chinese-to-English translation, source words are translated monotonously relative to their adjacent punctuation marks, which means their order relative to punctuation marks will not be changed. In summary, punctuation marks place a strong constraint on word order around them.

Chinese: 他 说 : 「 这 是 个 非常 严重 的 情况 , 我们 只能 希望 , 能 有 加快 行动 的 可能性 。 」
Bruin: urgent action , he said : "This is a very serious situation , we can only hope that there will be a possibility ."
Bruin+RH: he said : "This is a very serious situation , we can only hope that there will be the possibility to expedite action ."
Ref: He said: "This is a very serious situation. We can only hope that it is possible to speed up the operation."

Figure 1: An example of incorrect long-distance swap. The underlined Chinese words are incorrectly swapped to the beginning of the sentence by the original Bruin. RH means reordering heuristics.

For example, in Figure 1, Chinese words "加快行动" are reordered to sentence beginning. That is an incorrect long-distance swapping, which makes the reordered words moved out from the balanced punctuation marks "「" and "」", and incorrectly precede their previous mark "，".

These incorrect swappings definitely jeopardize the quality of translation. Here we propose two straightforward but effective heuristics to control and adjust the reordering, namely swapping window and punctuation restriction.

Swapping Window (SW): It constrains block swapping in the following way

$$\textbf{ACTIVATE } A \rightarrow \langle A^1, A^2 \rangle \textbf{ IF } |A^1_s| + |A^2_s| < sws$$

[1] In (Xiong et al., 2006), the language model uses the leftmost/rightmost words on the target side as evidences while the MaxEnt-based reordering model uses the boundary words on both sides.

where $|A_s^i|$ denotes the number of words on the source side A_s^i of block A^i, sws is a pre-defined swapping window size. Any inverted reordering beyond the pre-defined swapping window size is prohibited.

Punctuation Restriction (PR): If two neighboring blocks include any of the punctuation marks $p \in \{,\ 、\ :\ ;\ 「\ 」\ 《\ 》\ (\ )\ “\ ”\}$, the two blocks will be merged with straight order.

Punctuation marks were already used in parsing (Christine Doran, 2000) and statistical machine translation (Och et al., 2003). In (Och et al., 2003), three kinds of features are defined, all related to punctuation marks like quotes, parentheses and commas. Unfortunately, no statistically significant improvement on the BLEU score was reported in (Och et al., 2003). In this paper, we consider this problem from a different perspective. We emphasize that words around punctuation marks are reordered ungrammatically and therefore we positively use punctuation marks as a hard decision to restrict such reordering around punctuations. This is straightforward but yet results in significant improvement on translation quality.

The two heuristics described above can be used together. If the following conditions are satisfied, we can activate the inverted rule:

$$|A_s^1| + |A_s^2| < sws \;\&\&\; P \bigcap (A_s^1 \bigcup A_s^2) = \emptyset$$

where P is the set of punctuation marks mentioned above.

The two heuristics can also speed up decoding because decoding will be monotone within those spans which are not in accordance with both heuristics. For a sentence with n words, the total number of spans is $O(n^2)$. If we set $sws = m$ ($m < n$), then the number of spans with monotone search is $O((n-m)^2)$. With punctuation restriction, the non-monotone search space will reduce further.

3.2 Phrases with Sentence Beginning/Ending Tags

We observe that in a sentence some phrases are more likely to be located at the beginning, while other phrases are more likely to be at the end. This kind of location information with regard to the phrase position could be used for reordering. A straightforward way to use this information is to mark the beginning and ending of word-aligned sentences with ⟨s⟩ and ⟨/s⟩ respectively. This idea is borrowed from language modeling (Stolcke, 2002). The corresponding tags at the source and target sentences are aligned to each other, i.e, the beginning tag of source sentences is aligned to the beginning tag of target sentences, similarly for the ending tag. Figure 2 shows a word-aligned sentence pair annotated with the sentence beginning and ending tag.

During training, the sentence beginning and ending tags (⟨s⟩ and ⟨/s⟩) are treated as words. Therefore the phrase extraction and MaxEnt-based reordering training algorithm need not to be modified. Phrases with the sentence beginning/ending tag will be extracted and MaxEnt-based reordering features with such tags will also be generated. For example, from the word-aligned sentence pair in Figure 2, we can extract tagged phrases like

⟨s⟩ 西藏 ||| ⟨s⟩ Tibet 's

成绩 ⟨/s⟩ ||| achievements ⟨/s⟩

and generate MaxEnt-based reordering features with tags like

$$h_i(o, b^1, b^2) = \begin{cases} 1, & b^2.t_1 = \langle /s \rangle, o = s \\ 0, & otherwise \end{cases}$$

where b^1, b^2 are blocks, t_1 denotes the last source word, $o = s$ means the order between two blocks is straight. To avoid wrong alignments, we remove tagged phrases where only the beginning/ending tag is extracted on either side of the phrases, such as

⟨s⟩ ||| ⟨s⟩ Those 。

⟨/s⟩ ||| ⟨/s⟩

During decoding, we first annotate source sentences with the beiginning/ending tags, then translate them as what Bruin does. Note that phrases with sentence beginning/ending tags will be used in the same way as ordinary phrases without such tags during decoding. With the additional support of language model and MaxEnt-based reordering model, we observe that phrases with such tags are always moved to the beginning or ending of sentences correctly.

⟨s⟩	西藏	金融	工作	取得	显著	成绩	⟨/s⟩
⟨s⟩	Tibet 's	financial	work	has gained	remarkable	achievements	⟨/s⟩

Figure 2: A word-aligned sentence pair annotated with the sentence beginning and ending tag.

4 Evaluation

In this section, we report the performance of the enhanced Bruin on the NIST MT-05 and NIST MT-06 Chinese-to-English translation tasks. We describe the corpus, model training, and experiments related to the refinements described above.

4.1 Corpus

The bilingual training data is derived from the following various sources: the FBIS (LDC2003E14), Hong Kong Parallel Text (Hong Kong News and Hong Kong Hansards, LDC2004T08), Xinhua News (LDC2002E18), Chinese News Translation Text Part1 (LDC2005T06), Translations from the Chinese Treebank (LDC2003E07), Chinese English News Magazine (LDC2005E47). It contains 2.4M sentence pairs in total (68.1M Chinese words and 73.8M English words).

For the efficiency of minimum-error-rate training, we built our development set using sentences not exceeding 50 characters from the NIST MT-02 evaluation test data (580 sentences).

4.2 Training

We use exactly the same way and configuration described in (He et al., 2006) to preprocess the training data, align words and extract phrases.

We built two four-gram language models using Xinhua section of the English Gigaword corpus (181.1M words) and the English side of the bilingual training data described above respectively. We applied modified Kneser-Ney smoothing as implemented in the SRILM toolkit (Stolcke, 2002).

The MaxEnt-based reordering model is trained using the way of (Xiong et al., 2006). The difference is that we only use lexical features generated by tail words of blocks, instead of head words, removing features generated by the combination of two boundary words.

	Bleu(%)		Secs/sent	
Bruin	29.96		54.3	
sws	RH^1	RH_2^1	RH^1	RH_2^1
5	29.65	29.95	42.6	41.2
10	30.55	31.27	46.2	41.8
15	30.26	31.40	48.0	42.2
20	30.19	31.42	49.1	43.2

Table 1: Effect of reordering heuristics. RH^1 denotes swapping window while RH_2^1 denotes swapping window with the addition of punctuation restriction.

4.3 Translation Results

Table 1 compares the BLEU scores [2] and the speed in seconds/sentence of the baseline system Bruin and the enhanced system with reordering heuristics applied. The second row gives the BLEU score and the average decoding time of Bruin. The rows below row 3 show the BLEU scores and speed of the enhanced Bruin with different combinations of reordering heuristics. We can clearly see that the reordering heuristics proposed by us have a two-fold effect on the performance: improving the BLEU score and decreasing the average decoding time. The example in Figure 1 shows how reordering heuristics prevent incorrect long-distance swapping which is not in accordance with the punctuation restriction.

Table 1 also shows that a 15-word swapping window is an inflexion point with the best tradeoff between the decoding time and the BLEU score. We speculate that in our corpus most reorderings happen within a 15-word window. We use the FBIS corpus to testify this hypothesis. In this corpus, we extract all reordering examples using the algorithm of Xiong et al. (2006). Figure 3 shows the reordering length distribution curve in this corpus. Accord-

[2] In this paper, all BLEU scores are case-sensitive and evaluated on the NIST MT-05 Chinese-to-English translation task if there is no special note.

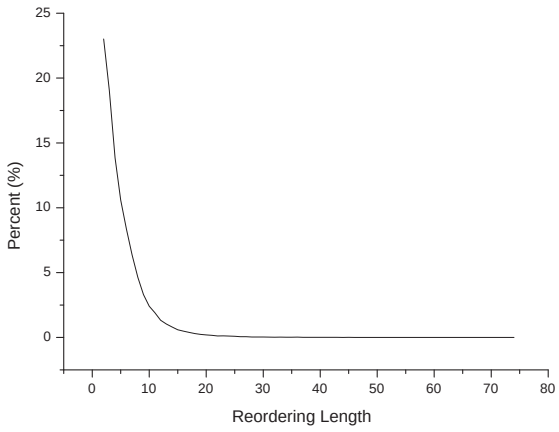


Figure 3: Reordering length distribution. The horizontal axis (reordering length) indicates the number of words on the source side of two neighboring blocks which are to be swapped. The vertical axis represents what proportion of reorderings with a certain length is likely to be in all reordering examples with an inverted order.

	Bleu(%)
Without Special Phrases	31.40
With Special Phrases	32.01

Table 2: Effect of integrating special phrases with the sentence beginning/ending tag.

ing to our statistics, reorderings within a window not exceeding 15 words have a very high proportion, 97.29%. Therefore we set $sws = 15$ for later experiments.

Table 2 shows the effect of integrating special phrases with sentence beginning/ending tags into Bruin. As special phrases accounts for only 1.95% of the total phrases used, an improvement of 0.6% in BLEU score is well worthwhile. Further, the improvement is statistically significant at the 99% confidence level according to Zhang's significant tester (Zhang et al., 2004). Figure 4 shows several examples translated with special phrases integrated. We can see that phrases with sentence beginning/ending tags are correctly selected and located at the right place.

Table 3 shows the performance of two systems on the NIST MT-05 Chinese test data, which are (1) Bruin, trained on the large data described above; and (2) enhanced Bruin (EBruin) with different refinements trained on the same data set. This table also shows the evaluation result of the enhanced Bruin with reordering heuristics, obtained in the NIST MT-06 evaluation exercise. [3]

System	Refine	MT-05	MT-06
Bruin	-	29.96	-
EBruin	RH	31.40	30.22
EBruin	RH+SP	32.01	-

Table 3: Results of different systems. The refinements RH, SP represent reordering heuristics and special phrases with the sentence beginning/ending tag, respectively.

5 Conclusions

We have described in detail two refinements for BTG-based SMT which include reordering heuristics and special phrases with tags. The refinements were integrated into a well-established BTG-based system Bruin introduced by Xiong et al. (2006). Reordering heuristics proposed here achieve a twofold improvement: better reordering and higher-speed decoding. To our best knowledge, we are the first to integrate special phrases with the sentence beginning/ending tag into SMT. Experimental results show that the above refinements improve the baseline system significantly.

For further improvements, we will investigate possible extensions to the BTG grammars, e.g. learning useful nonterminals using unsupervised learning algorithm.

Acknowledgements

We would like to thank the anonymous reviewers for useful comments on the earlier version of this paper. The first author was partially supported by the National Science Foundations of China (No. 60573188) and the High Technology Research and Development Program of China (No. 2006AA010108) while he studied in the Institute of Computing Technology, Chinese Academy of Sciences.

[3]Full results are available at http://www.nist.gov/speech/tests/mt/doc/mt06eval_official_results.html.

With Special Phrases	Without Special Phrases
⟨s⟩ **Japan had** already pledged to provide 30 million US dollars of aid due to the tsunami victims of the country . ⟨/s⟩	**originally has** pledged to provide 30 million US dollars of aid from Japan tsunami victimized countries .
⟨s⟩ **the results of the survey is based on** the results of the chiefs of the Ukrainian National 50.96% cast by chiefs . ⟨/s⟩	**is based on** the survey findings Ukraine 50.96% cast by the chiefs of the chiefs of the country .
⟨s⟩ and at the same time , the focus of the world have been transferred to **other areas . ⟨/s⟩**	and at the same time , **the global focus has shifted he**.

Figure 4: Examples translated with special phrases integrated. The bold underlined words are special phrases with the sentence beginning/ending tag.

References

Yaser Al-Onaizan, Kishore Papineni. 2006. Distortion Models for Statistical Machine Translation. In *Proceedings of ACL-COLING 2006*.

David Chiang, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, Michael Subotin. 2005. The Hiero Machine Translation System: Extensions, Evaluation, and Analysis. In *Proceedings of HLT/EMNLP*, pages 779 – 786, Vancouver, October 2005.

David Chiang. 2007. Hierarchical Phrase-based Translation. In *computational linguistics*, 33(2).

Christine Doran. 2000. Punctuation in a Lexicalized Grammar. In *Proceedings of Workshop TAG+5*, Paris.

Zhongjun He, Yang Liu, Deyi Xiong, Hongxu Hou, Qun Liu. 2006. ICT System Description for the 2006 TC-STAR Run #2 SLT Evaluation. In *Proceedings of TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain.

Liang Huang, Hao Zhang and Daniel Gildea. 2005. Machine Translation as Lexicalized Parsing with Hooks. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT-05)*, Vancouver, BC, Canada, October 2005.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, pages 115 – 124.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *International Workshop on Spoken Language Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. ACL 2007, demonstration session, Prague, Czech Republic, June 2007.

Yang Liu, Qun Liu, Shouxun Lin. 2006. Tree-to-String Alignment Template for Statistical Machine Translation. In *Proceedings of ACL-COLING 2006*.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of CoNLL-2002*.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical Machine Translation with Syntactified Target Language Phraases. In *Proceedings of EMNLP*.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL 2002*, pages 295 – 302.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, Dragomir Radev. 2003. Final Report of Johns Hopkins 2003 Summer Workshop on Syntax for Statistical Machine Translation.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, volume 2, pages 901-904.

Dekai Wu. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proceedings of IJCAL 1995*, pages 1328-1334, Montreal, August.

Dekai Wu. 1996. A Polynomial-Time Algorithm for Statistical Machine Translation. In *Proceedings of ACL 1996*.

Deyi Xiong, Qun Liu and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proceedings of ACL-COLING 2006*, pages 521 – 528.

R. Zens, H. Ney, T. Watanabe, and E. Sumita. 2004. Reordering Constraints for Phrase-Based Statistical Machine Translation. In *Proceedings of CoLing 2004*, Geneva, Switzerland, pp. 205-211.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Proceedings of LREC 2004*, pages 2051 – 2054.

Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation

Ananthakrishnan Ramanathan, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology
Powai, Mumbai-400076
India
{anand,pb}@cse.iitb.ac.in

Jayprasad Hegde, Ritesh M. Shah, Sasikumar M
CDAC Mumbai (formerly NCST)
Gulmohar Cross Road No. 9
Juhu, Mumbai-400049
India
{jjhegde,ritesh,sasi}
@cdacmumbai.in

Abstract

In this paper, we report our work on incorporating syntactic and morphological information for English to Hindi statistical machine translation. Two simple and computationally inexpensive ideas have proven to be surprisingly effective: (i) reordering the English source sentence as per Hindi syntax, and (ii) using the suffixes of Hindi words. The former is done by applying simple transformation rules on the English parse tree. The latter, by using a simple suffix separation program. With only a small amount of bilingual training data and limited tools for Hindi, we achieve reasonable performance and substantial improvements over the baseline phrase-based system. Our approach eschews the use of parsing or other sophisticated linguistic tools for the target language (Hindi) making it a useful framework for statistical machine translation from English to Indian languages in general, since such tools are not widely available for Indian languages currently.

1 Introduction

Techniques for leveraging syntactic and morphological information for statistical machine translation (SMT) are receiving a fair amount of attention nowadays. For SMT from English to Indian languages, these techniques are especially important for the following three reasons: (i) Indian languages differ widely from English in terms of word-order; (ii) Indian languages are morphologically quite rich; and (iii) large amounts of parallel corpora are not available for these languages, though smaller amounts of text in specific domains (such as health, tourism, and agriculture) are now becoming accessible. It might therefore be expected that using syntactic and morphological information for English to Indian language SMT will prove highly beneficial in terms of achieving reasonable performance out of limited parallel corpora. However, the difficulty in this is that crucial tools, such as parsers and morphological analyzers, are not widely available for Indian languages yet.

In this paper, we present our work on incorporating syntactic and morphological information for English to Hindi SMT. Our approach, which eschews the use of parsing and other tools for Hindi, is two-pronged:

1. Incorporating syntactic information by combining phrase-based models with a set of structural preprocessing rules on English
2. Incorporating morphological information by using a simple suffix separation program for Hindi, the likes of which can be created with limited effort for other Indian languages as well

Significant improvements over the baseline phrase-based SMT system are obtained using our approach. Table 1 illustrates this with an example [1].

Since only limited linguistic effort and tools are required for the target language, we believe that the framework we propose is suitable for SMT from English to other Indian languages as well.

[1] This example is discussed further in section 4

input	For a celestial trip of the scientific kind, visit the planetarium.
reference	वैग्यानिक तरीके के एक दिव्य सैर के लिए, तारामंडल आएं। vaigyaanika tariike ke eka divya saira ke lie, taaraamandala aaem *scientific kind of a celestial trip for, planetarium visit (come)*
baseline	के स्वर्गीय यात्रा के वैग्यानिक प्रकार, का तारागृह है। ke svargiiya yaatraa ke vaigyaanika prakaara, kaa taaraagruha hai *of celestial trip of scientific kind, of planetarium is*
baseline+syn	वैग्यानिक प्रकार के स्वर्गीय यात्रा के लिए, तारागृह है। vaigyaanika prakaara ke svargiiya yaatraa ke lie, taaraagruha hai *scientific kind of celestial trip for, planetarium is*
baseline+syn+morph	वैग्यानिक प्रकार के स्वर्गीय यात्रा के लिए, तारागृह देखें। vaigyaanika prakaara ke svargiiya yaatraa ke lie, taaraagruha dekhem *scientific kind of celestial trip for, planetarium visit (see)*

Table 1: **Effects of Syntactic and Morphological Processing** (*reference*: human reference translation; *baseline*: phrase-based system; *syn*: with syntactic information; *morph*: with morphological information)

The rest of this paper is organized as follows: Section 2 outlines related work. Section 3 describes our approach – first, the phrase-based baseline system is sketched briefly, leading up to the techniques used for incorporating syntactic and morphological information within this system. Experimental results are discussed in section 4. Section 5 concludes the paper with some directions for future work.

2 Related Work

Statistical translation models have evolved from the word-based models originally proposed by Brown et al. (1990) to syntax-based and phrase-based techniques.

The beginnings of phrase-based translation can be seen in the alignment template model introduced by Och et al. (1999). A joint probability model for phrase translation was proposed by Marcu and Wong (2002). Koehn et al. (2003) propose certain heuristics to extract phrases that are consistent with bidirectional word-alignments generated by the IBM models (Brown et al., 1990). Phrases extracted using these heuristics are also shown to perform better than syntactically motivated phrases, the joint model, and IBM model 4 (Koehn et al., 2003).

Syntax-based models use parse-tree representations of the sentences in the training data to learn, among other things, tree transformation probabilities. These methods require a parser for the target language and, in some cases, the source language too. Yamada and Knight (2001) propose a model that transforms target language parse trees to source language strings by applying reordering, insertion, and translation operations at each node of the tree. Graehl and Knight (2004) and Melamed (2004), propose methods based on tree-to-tree mappings. Imamura et al. (2005) present a similar method that achieves significant improvements over a phrase-based baseline model for Japanese-English translation.

Recently, various preprocessing approaches have been proposed for handling syntax within SMT. These algorithms attempt to reconcile the word-order differences between the source and target language sentences by reordering the source language data prior to the SMT training and decoding cycles. Nießen and Ney (2004) propose some restructuring steps for German-English SMT. Popovic and Ney (2006) report the use of simple local transformation rules for Spanish-English and Serbian-English translation. Collins et al. (2006) propose German clause restructuring to improve German-English SMT.

The use of morphological information for SMT has been reported in (Nießen and Ney, 2004) and (Popovic and Ney, 2006). The detailed experiments by Nießen and Ney (2004) show that the use of morpho-syntactic information drastically reduces the need for bilingual training data.

Recent work by Koehn and Hoang (2007) pro-

poses factored translation models that combine feature functions to handle syntactic, morphological, and other linguistic information in a log-linear model.

Our work uses a preprocessing approach for incorporating syntactic information within a phrase-based SMT system. For incorporating morphology, we use a simple suffix removal program for Hindi and a morphological analyzer for English. These aspects are described in detail in the next section.

3 Syntactic & Morphological Information for English-Hindi SMT

3.1 Phrase-Based SMT: the Baseline

Given a source sentence f, SMT chooses as its translation $\hat{e}$, which is the sentence with the highest probability:

$$\hat{e} = \underset{e}{arg\,max}\ p(e|f)$$

According to Bayes' decision rule, this is written as:

$$\hat{e} = \underset{e}{arg\,max}\ p(e)p(f|e)$$

The phrase-based model that we use as our baseline system (defined by Koehn et al. (2003)) computes the translation model $p(f|e)$ by using a phrase translation probability distribution. The decoding process works by segmenting the input sentence f into a sequence of I phrases $\overline{f}_1^I$. A uniform probability distribution over all possible segmentations is assumed. Each phrase $\overline{f}_i$ is translated into a target language phrase $\overline{e}_i$ with probability $\phi(\overline{f}_i|\overline{e}_i)$. Reordering is penalized according to a simple exponential distortion model.

The phrase translation table is learnt in the following manner: The parallel corpus is word-aligned bidirectionally, and using various heuristics (see (Koehn et al., 2003) for details) phrase correspondences are established. Given the set of collected phrase pairs, the phrase translation probability is calculated by relative frequency:

$$\phi(\overline{f}|\overline{e}) = \frac{count(\overline{f}, \overline{e})}{\sum_f count(f, \overline{e})}$$

Lexical weighting, which measures how well words within phrase pairs translate to each other, validates the phrase translation, and addresses the problem of data sparsity.

The language model $p(e)$ used in our baseline system is a trigram model with modified Kneser-Ney smoothing (Chen and Goodman, 1998).

The weights for the various components of the model (phrase translation model, language model, distortion model etc.) are set by minimum error rate training (Och, 2003).

3.2 Syntactic Information

As mentioned in section 2, phrase-based models have emerged as the most successful method for SMT. These models, however, do not handle syntax in a natural way. Reordering of phrases during translation is typically managed by distortion models, which have proved not entirely satisfactory (Collins et al., 2006), especially for language pairs that differ a lot in terms of word-order. We use a preprocessing approach to get over this problem, by reordering the English sentences in the training and test corpora before the SMT system kicks in. This reduces, and often eliminates, the 'distortion load' on the phrase-based system.

The reordering rules that we use for preprocessing can be broadly described by the following transformation rule going from English to Hindi word order (Rao et al, 2000):

$$SS_mVV_mOO_mCm \rightarrow C'_mS'_mS'O'_mO'V'_mV'$$

where,
S: Subject
O: Object
V: Verb
C_m: Clause modifier
X': Corresponding constituent in Hindi, where X is S, O, or V
X_m: modifier of X

Essentially, the SVO order of English is changed to SOV order, and post-modifiers are converted to pre-modifiers. Our preprocessing module effects this by parsing the input English sentence [2] and ap-

[2] Dan Bikel's parser was used for parsing (http://www.cis.upenn.edu/dbikel/license.html).

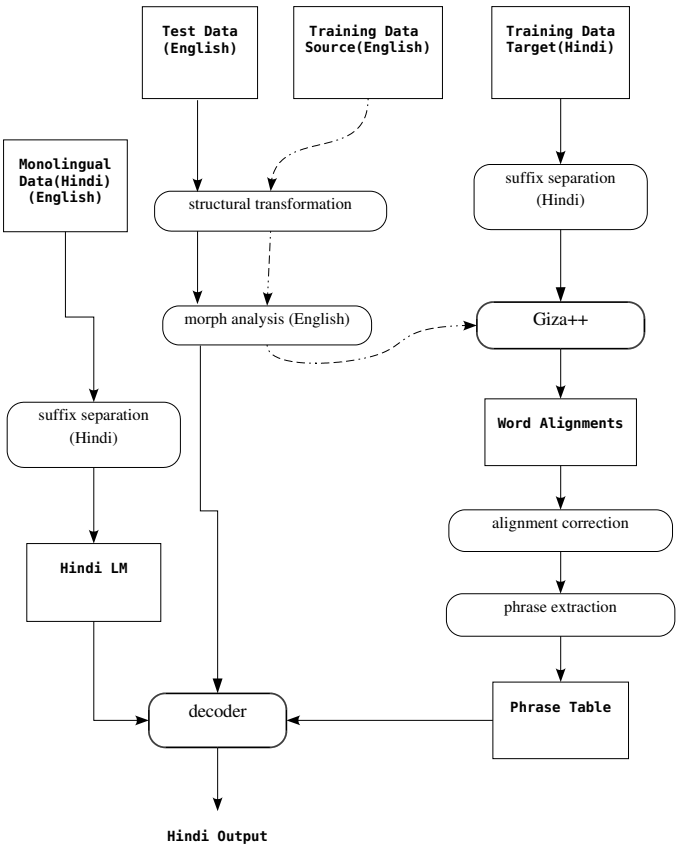


Figure 1: **Syntactic and Morphological Processing: Schematic**

plying a handful of reordering rules on the parse tree. Table 2 illustrates this with an example.

3.3 Morphological Information

If an SMT system considers different morphological forms of a word as independent entities, a crucial source of information is neglected. It is conceivable that with the use of morphological information, especially for morphologically rich languages, the requirement for training data might be much reduced. This is indicated, for example, in recent work on German-English statistical MT with limited bilingual training data (Nießen and Ney, 2004), and also in other applications such as statistical part-of-speech tagging of Hindi (Gupta et al., 2006).

The separation of morphological suffixes conflates various forms of a word, which results in higher counts for both words and suffixes, thereby countering the problem of data sparsity. As an example, assume that the following sentence pair is part of the bilingual training corpus:

> **English**: Players should just play.
> **Hindi**: खिलाडियों को केवल खेलना चाहिए।
> *khilaadiyom ko kevala khelanaa caahie*
> **Hindi (suffix separated)**: खिलाड इयों को केवल खेल ना चाहिए।
> *khilaada iyom ko kevala khela naa caahie*

Now, consider the input sentence, "The men came across some players," which should be translated as "आदमियों को कुछ खिलाडी मिले" (*aadmiyom ko kucha khilaadii mile*). Without using morphology, the system is constrained to the choice of खिलाडियों (*khilaadiyom*) for the word *players* (based just on the

English	$\overbrace{The\ president}^{S}\ \overbrace{of\ America}^{S_m}\ \overbrace{visited}^{V}\ \overbrace{India}^{O}\ \overbrace{in\ June}^{V_m}$
Reordered	$\underbrace{America\ of}_{S_m}\ \underbrace{the\ president}_{S}\ \underbrace{June\ in}_{V_m}\ \underbrace{India}_{O}\ \underbrace{visited}_{V}$
Hindi	अमरीका के राष्ट्रपति ने जून में भारत की यात्रा की amariikaa ke raashtrapati ne juuna mem bhaarata kii yaatraa kii

Table 2: **English and Hindi Word-Order**

आ	आएं	अता	आने	एगा
इ	आओं	अती	ऊंगा	एगी
ई	इयां	ईं	ऊंगी	आएगा
उ	इयों	अतिं	आऊंगा	आएगी
ऊ	आइयां	अते	आऊंगी	आया
ए	आइयों	आता	एंगे	आए
ओ	आँ	आती	एंगी	आई
एं	इयाँ	आतीं	आएंगे	आईं
ओं	आइयाँ	आते	आएंगी	इए
आं	अताएं	अना	ओगे	आओ
उआं	अताओं	अनी	ओगी	आइए
उएं	अनाएं	अने	आओगे	अकर
उओं	अनाओं	आना	आओगी	आकर

Table 3: **Hindi Suffix List**

evidence from the above sentence pair in the training corpus). Also, the general relationship between the oblique case (indicated by the suffix इयों (*iyom*)) and the case marker को (*ko*) is not learnt, but only the specific relationship between खिलाडियों (*khilaadiyom*) and को (*ko*). This indicates the necessity of using morphological information for languages such as Hindi.

To incorporate morphological information, we use a morphological analyzer (Minnen et al., 2001) for English, and a simple suffix separation program for Hindi. The suffix separation program is based on the Hindi stemmer presented in (Ananthakrishnan and Rao, 2003), and works by separating from each word the longest possible suffix from table 3. A detailed analysis of noun, adjective, and verb inflections that were used to create this list can be found in (McGregor, 1977) and (Rao, 1996). A few examples of each type are given below:

Noun Inflections: Nouns in Hindi are inflected based on the case (direct or oblique), the number (singular or plural), and the gender (masculine or feminine[3]). For example, लडका (*ladakaa* - boy) becomes लडके (ladake) when in oblique case, and the plural लडके (*ladake* - boys) becomes लडकों (*ladakom*). The feminine noun लडकी (*ladakii* - girl) is inflected as लडकियाँ (*ladakiyaam* - plural direct) and लडकियों (*ladakiyom* - plural oblique), but it remains uninflected in the singular direct case.

Adjective Inflections: Adjectives which end in आ (*aa*) or आं (*aam*) in their direct singular masculine form agree with the noun in gender, number, and case. For example, the singular direct अच्छा (*accha*) is inflected as अच्छे (*acche*) in all other masculine forms, and as अच्छी (*acchii*) in all feminine forms. Other adjectives are not inflected.

Verb Inflections: Hindi verbs are inflected based on gender, number, person, tense, aspect, modality, formality, and voice. (Rao, 1996) provides a complete list of verb inflection rules.

The overall process used for incorporating syntactic and morphological information, as described in this section, is shown in figure 1.

[3]Hindi does not possess a neuter gender

Technique	Evaluation Metric				
	BLEU	mWER	SSER	roughly understandable+	understandable+
baseline	12.10	77.49	91.20	10%	0%
baseline+syn	16.90	69.18	74.40	42%	12%
baseline+syn+morph	15.88	70.69	66.40	46%	28%

Table 4: **Evaluation Results** (*baseline*: phrase-based system; *syn*: with syntactic information; *morph*: with morphological information)

4 Experimental Results

The corpus described in the table below was used for the experiments.

	#**sentences**	#**words**
Training	5000	120,153
Development	483	11,675
Test	400	8557
Monolingual (Hindi)	49,937	1,123,966

The baseline system was implemented by training the phrase-based system described in section 3 on the 5000 sentence training corpus.

For the Hindi language model, we compared various n-gram models, and found trigram models with modified Kneser-Ney smoothing to be the best performing (Chen and Goodman, 1998). One language model was learnt from the Hindi part of the 5000 sentence training corpus. The larger monolingual Hindi corpus was used to learn another language model. The SRILM toolkit [4] was used for the language modeling experiments.

The development corpus was used to set weights for the language models, the distortion model, the phrase translation model etc. using minimum error rate training. Decoding was performed using Pharaoh [5].

fnTBL (Ngai and Florian, 2001) was used to POS tag the English corpus, and Bikel's parser was used for parsing. The reordering program was written using the perl module Parse::RecDescent.

We evaluated the various techniques on the following criteria. For the objective criteria (BLEU and mWER), two reference translations per sentence were used.

- **BLEU** (Papineni et al., 2001): This measures the precision of n-grams with respect to the reference translations, with a brevity penalty. A higher BLEU score indicates better translation.
- **mWER** (multi-reference word error rate) (Nießen et al., 2000): This measures the edit distance with the most similar reference translation. Thus, a lower mWER score is desirable.
- **SSER** (subjective sentence error rate) (Nießen et al., 2000): This is calculated using human judgements. Each sentence was judged by a human evaluator on the following five-point scale, and the SSER was calculated as described in (Nießen et al., 2000).

0	Nonsense
1	Roughly understandable
2	Understandable
3	Good
4	Perfect

  Again, the lower the SSER, the better the translation.

Table 4 shows the results of the evaluation. We find that using syntactic preprocessing brings substantial improvements over the baseline phrase-based system. While the impact of morphological information is not seen in the BLEU and mWER scores, the subjective scores reveal the effectiveness of using morphology. The last two columns of the table show the percentage of sentences that were found by the human judges to be roughly understandable (or higher) and understandable (or higher) respectively in the evaluation scale. We find that including syntactic and morphological information brings substantial improvements in translation fluency.

[4] http://www.speech.sri.com/projects/srilm/
[5] http://www.isi.edu/licensed-sw/pharaoh/

An Example: Consider, again, the example in table 1. The word-order in the baseline translation is woeful, while the translations after syntactic preprocessing (baseline+syn and baseline+syn+morph) follow the correct Hindi order (compare with the reference translation). The effect of suffix separation can be seen from the verb form (देखें (*dekhem*) – visit or see) in the last translation (baseline+syn+morph). The reason for this is that the pair "visit → देखें" is not available to be learnt from the original and the syntactically preprocessed corpora, but the following pairs are: (i) to visit → देखना (ii) worth visiting → देखने योग्य, and (iii) can visit → देख सकते हैं. Thus, the baseline and baseline+syn models are not able to produce the correct verb form for "visit". On the other hand, the baseline+syn+morph model, due to the suffix separation process, combines देख (*dekha*) and एं (*em*) from different mappings in the aligned corpus, e.g., "visit +ing → देख ने" and "sing → गा एं", to get the right translation for visit (देखें) in this context.

5 Conclusion

We have presented in this paper an effective framework for English-Hindi phrase-based SMT. The results demonstrate that significant improvements are possible through the use of relatively simple techniques for incorporating syntactic and morphological information.

Since all Indian languages follow SOV order, and are relatively rich in terms of morphology, the framework presented should be applicable to English to Indian language SMT in general. Given that morphological and parsing tools are not yet widely available for Indian languages, an approach like ours which minimizes use of such tools for the target language would be quite desirable.

In future work, we propose to experiment with a more sophisticated morphological analyzer. As more parallel corpora become available, we also intend to measure the effects of using morphology on corpora requirements. Finally, a formal evaluation of these techniques for other Indian languages (especially Dravidian languages such as Tamil) would be interesting.

Acknowledgements

We are grateful to Sachin Anklekar and Saurabh Kushwaha for their assistance with the tedious task of collecting and preprocessing the corpora.

References

Ananthakrishnan Ramanathan and Durgesh Rao, A Lightweight Stemmer for Hindi, *Workshop on Computational Linguistics for South-Asian Languages*, EACL, 2003.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, A Statistical Approach to Machine Translation, *Computational Linguistics*, 16(2), pages 79–85, June 1990.

Stanley F. Chen and Joshua T. Goodman, An Empirical Study of Smoothing Techniques for Language Modeling, *Technical Report TR-10-98*, Computer Science Group, Harvard University, 1998.

Michael Collins, Philipp Koehn, and Ivona Kucerova, Clause Restructuring for Statistical Machine Translation, *Proceedings of ACL*, pages 531–540, 2006.

Jonathan Graehl and Kevin Knight, Training Tree Transducers, *Proceedings of HLT-NAACL*, 2004.

Kuhoo Gupta, Manish Shrivastava, Smriti Singh, and Pushpak Bhattacharyya, Morphological Richness Offsets Resource Poverty – an Experience in Builing a POS Tagger for Hindi, *Proceedings of ACL-COLING*, 2006.

Kenji Imamura, Hideo Okuma, Eiichiro Sumita, Practical Approach to Syntax-based Statistical Machine Translation, *Proceedings of MT-SUMMIT X*, 2005.

Philipp Koehn and Hieu Hoang, Factored Translation Models, *Proceedings of EMNLP*, 2007.

Philip Koehn, Franz Josef Och, and Daniel Marcu, Statistical Phrase-based Translation, *Proceedings of HLT-NAACL*, 2003.

Daniel Marcu and William Wong, A Phrase-based Joint Probability Model for Statistical Machine Translation, *Proceedings of EMNLP*, 2002.

R. S. McGregor, *Outline of Hindi Grammar*, Oxford University Press, Delhi, India, 1974.

I. Dan Melamed, Statistical Machine Translation by Parsing, *Proceedings of ACL*, 2004.

Guido Minnen, John Carroll, and Darren Pearce, Applied Morphological Processing of English, *Natural Language Engineering*, 7(3), 207–223, 2001.

G. Ngai and R. Florian, Transformation-based Learning in the Fast Lane, *Proceedings of NAACL*, 2001.

Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney, An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research, *International Conference on Language Resources and Evaluation*, pages 39–45, 2000.

Sonja Nießen and Hermann Ney, Statistical Machine Translation with Scarce Resources Using Morpho-syntactic Information, *Computational Linguistics*, 30(2), pages 181–204, 2004.

Franz Josef Och, Christoph Tillman, and Hermann Ney, Improved Alignment Models for Statistical Machine Translation, *Proceedings of EMNLP*, pages 20–28, 1999.

Franz Josef Och, Minimum Error Rate Training in Statistical Machine Translation, *Proceedings of ACL*, 2003.

Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, *IBM Research Report*, Thomas J. Watson Research Center, 2001.

Maja Popovic and Hermann Ney, Statistical Machine Translation with a Small Amount of Bilingual Training Data, *5th LREC SALTMIL Workshop on Minority Languages*, pages 25–29, 2006.

Durgesh Rao, Natural Language Generation for English to Hindi Human-Aided Machine Translation of News Stories, *Master's Thesis*, Indian Institute of Technology, Bombay, 1996.

Durgesh Rao, Kavitha Mohanraj, Jayprasad Hegde, Vivek Mehta, and Parag Mahadane, A Practical Framework for Syntactic Transfer of Compound-Complex Sentences for English-Hindi Machine Translation, *Proceedings of KBCS*, 2000.

Kenji Yamada and Kevin Knight A Syntax-based Statistical Translation Model, *Proceedings of ACL*, 2001.

Statistical Machine Translation Models for Personalized Search

Rohini U *
AOL India R& D
Bangalore, India
Rohini.uppuluri@corp.aol.com

Vamshi Ambati
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, USA
vamshi@cs.cmu.edu

Vasudeva Varma
LTRC, IIIT Hyd
Hyderabad, India
vv@iiit.ac.in

Abstract

Web search personalization has been well studied in the recent few years. Relevance feedback has been used in various ways to improve relevance of search results. In this paper, we propose a novel usage of relevance feedback to effectively model the process of query formulation and better characterize how a user relates his query to the document that he intends to retrieve using a noisy channel model. We model a user profile as the probabilities of translation of query to document in this noisy channel using the relevance feedback obtained from the user. The user profile thus learnt is applied in a re-ranking phase to rescore the search results retrieved using an underlying search engine. We evaluate our approach by conducting experiments using relevance feedback data collected from users using a popular search engine. The results have shown improvement over baseline, proving that our approach can be applied to personalization of web search. The experiments have also resulted in some valuable observations that learning these user profiles using snippets surrounding the results for a query gives better performance than learning from entire document collection.

1 Introduction

Most existing text retrieval systems, including the web search engines, suffer from the problem of "one size fits all": the decision of which documents to retrieve is made based only on the query posed, without consideration of a particular user's preferences and search context. When a query (e.g. "jaguar") is ambiguous, the search results are inevitably mixed in content (e.g. containing documents on the jaguar cat and on the jaguar car), which is certainly non-optimal for a given user, who is burdened by having to sift through the mixed results. In order to optimize retrieval accuracy, we clearly need to model the user appropriately and personalize search according to each individual user. The major goal of personalized search is to accurately model a user's information need and store it in the user profile and then re-rank the results to suit to the user's interests using the user profile. However, understanding a user's information need is, unfortunately, a very difficult task partly because it is difficult to model the search process which is a cognitive process and partly because it is difficult to characterize a user and his preferences and goals. Indeed, this has been recognized as a major challenge in information retrieval research (et. al, 2003).

In order to address the problem of personalization one needs to clearly understand the actual process of search. First the user has an information need that he would like to fulfill. He is the only entity in the process that knows the exact information he needs and also has a vague notion of the document that can full fill his specific information need. A query based search engine is at his disposal for identifying this particular document or set of documents from among a vast repository of them. He then formulates a query that he thinks is congruent to the document he imagines to fulfill his need and poses it to the search engine. The search engine now returns

*This work was done when the first and second authors were at IIIT Hyderabad, India.

a list of results that it calculates as relevant according to its ranking algorithm. Every user is different and has a different information need, perhaps overlapping sometimes. The way a user conceives an ideal document that fulfills his need also varies. It is our hypothesis that if one can learn the variations of each user in this direction, effective personalization can be done.

Most approaches to personalization have tried to model the user's interests by requesting explicit feedback from the user during the search process and observing these relevance judgments to model the user's interests. This is called relevance feedback, and personalization techniques using it have been proven to be quite effective for improving retrieval accuracy (Salton and Buckley, 1990; Rocchio, 1971). These approaches to personalization have considered, user profile to be a collection of words, ontology, a matrix etc.

We use relevance feedback for personalization in our approach. However we propose a novel usage of relevance feedback to effectively model the process of query formulation and better characterize how a user relates his query to the document that he intends to retrieve as discussed in the web search process above. A user profile learnt from the relevance feedback that captures the query generation process is used as a guide to understand user's interests over time and personalize his web search results.

Interestingly, a new paradigm has been proposed for retrieval rooted from statistical language modeling recently that views the query generation process through a Noisy channel model (Berger and Lafferty, 1999) . It was assumed that the document and query are from different languages and the query generation process was viewed as a translation from the document language which is more verbose to the language of the query which is more compact and brief. The noisy channel model proposed by Berger and Lafferty (Berger and Lafferty, 1999) inherently captures the dependencies between the query and document words by learning a translation model between them. As we intend to achieve personalized search by personalizing the query formulation process, we also perceive the user profile learning through a Noisy Channel Model. In the model, when a user has an information need, he also has an ideal document in mind that fulfills his need. The user tries to in a way translate the notion of the ideal document into a query that is more compact but congruent to the document. He then poses this query to the search engine and retrieves the results. By observing this above process over time, we can capture how the user is generating a query from his ideal document. By learning this model of a user, we can predict which document best describes his information need for the query he poses. This is the motive of personalization. In our approach, we learn a user model which is probabilistic model for the noisy channel using statistical translation approaches and from the past queries and their corresponding relevant documents provided as feedback by the user.

The rest of the paper is organized as follows. We first describe the related work on personalized search then we provide the background and the framework that our approach is based upon. we discuss the modeling of a user profile as a translation model. after which we describe applying it to personalized search. we describe our experimental results followed by conclusions with directions to some future work.

2 Related Work

There has been a growing literature available with regard to personalization of search results. In this section, we briefly overview some of the available literature.

(Pretschner and Gauch, 1999) used ontology to model users interests, which are studied from users browsed web pages. (Speretta and Gauch, 2004) used users search history to construct user profiles. (Liu et al., 2002) performed personalized web search by mapping a query to a set of categories using a user profile and a general profile learned from the user's search history and a category hierarchy respectively. (Hatano and Yoshikawa., 2004) considered the unseen factors of the relationship between the web users behaviors and information needs and constructs user profiles through a memory-based collaborative filtering approach.

To our knowledge, there has been a very little work has been done that explicitly uses language models to personalization of search results. (Croft et al., 2001) discuss about relevance feedback and

query expansion using language modeling. (Shen et al., 2005) use language modeling for short term personalization by expanding queries.

Earlier approaches to personalization have considered, user profile to be a collection of words, ontology, language model etc. We perceive the user profile learning through a Noisy Channel Model. In the model, when a user has an information need, he also has a vague notion of what is the ideal document that he would like to retrieve. The user then creates a compact query that he thinks would retrieve the document. He then poses the query to the search engine. By observing this above process over time, we learn a user profile as the probabilities of translation for the noisy channel that converts his document to the query. We then use this profile in re-ranking the results of a search engine to provide personalized results.

3 Background

In this section, we describe the statistical language modeling and the translation model framework for information retrieval that form a basis for our research.

The basic approach for language modeling for IR was proposed by Ponte and Croft (Ponte and Croft, 1998). It assumes that the user has a reasonable idea of the terms that are likely to appear in the ideal document that can satisfy his/her information need, and that the query terms the user chooses can distinguish the ideal document from the rest of the collection. The query is thus generated as the piece of text representative of the ideal document. The task of the system is then to estimate, for each of the documents in the collection, which is most likely to be the ideal document.

$$\arg\max_D P(D|Q) = \arg\max_D P(Q|D)P(D)$$

where Q is a query and D is a document. The prior probability $P(D)$ is usually assumed to be uniform and a language model $P(Q|D)$ is estimated for every document. In other words, they estimate a probability distribution over words for each document and calculate the probability that the query is a sample from that distribution. Documents are ranked according to this probability. The basic model has been extended in a variety of ways. Modeling documents as in terms of a noisy channel model by Berger & Lafferty (Berger and Lafferty, 1999), mixture of topics, and phrases are considered (Song and Croft., 1999), (Lavrenko and Croft, 2001) explicitly models relevance, and a risk minimization framework based on Bayesian decision theory has been developed (Zhai and Lafferty, 2001).

The noisy channel by Berger and Lafferty (Berger and Lafferty, 1999) view a query as a distillation or translation from a document describing the query generation process in terms of a noisy channel model. In formulating a query to a retrieval system, a user begins with an information need. This information need is then represented as a fragment of an "ideal document", a portion of the type of document that the user hopes to receive from the system. The user then translates or "distills" this ideal document fragment into a succinct query, selecting key terms and replacing some terms with related terms.

To determine the relevance of a document to a query, their model estimates the probability that the query would have been generated as a translation of that document. Documents are then ranked according to these probabilities. More specifically, the mapping from a document term w to a query term q_i is achieved by estimating translation models $P(q|w)$. Using translation models, the retrieval model becomes

$$P(Q|D) = \prod_{q_i \in Q} \alpha P(q_i|GE) + (1-\alpha) \sum_{w \in D} P(q_i|w)P(w|D)$$

where $P(q_i|GE)$ is the smoothed or general probability obtained from a large general corpus. $P(q_i|w)$ is an entry in the translation model. It represents the probability of generation of the query word q_i for a word w in the document. $P(w|D)$ is the probability of the word w in the document and α is a weighting parameter which lies between 0 and 1.

4 User Profile as a Translation Model

We perceive the user profile learning as learning the channel probabilities of a Noisy Channel Model that generates the query from the document. In the model, when a user has an information need, he also has a vague notion of what is the ideal document that he would like to retrieve. The user then creates

a compact query that he thinks would retrieve the document. He then poses the query to the search engine. By observing this above process over time, we can learn how the user is generating a query from his notion of an ideal document. By learning this, we can predict which document best describes his information need. The learnt model, called a user profile, is thus capable of personalizing results for that particular user. Hence, the user profile here is a translation model learnt from explicit feedback of the user using statistical translation approaches. Explicit feedback consists of the past queries and their corresponding relevant documents provided as feedback by the user. A translation model is a probabilistic model consisting of the triples, the source word, the target word and the probability of translation. The translation model here is between document words and queries words. Therefore the user profile as a translation model in our approach will consist of triples of a document word, a query word and the probability of the document word generating the query word.

5 Personalized Search

In this section, we describe how we perform personalized search using the proposed translation model based user profile. First, a user profile is learnt using the translation model process then the re-ranking is done using the learnt user profile.

5.1 Learning user profile

In our approach, a user profile consists of a statistical translation model. A translation model is a probabilistic model consisting of the triples, the source word, the target word and the probability of translation. Our user profiles consists of the following triples, a document word, a query word and the probability of the document word generating the query word.

Consider a user u, let $\{ \{Q_i, D_i\}, i = 1, 2, ..., N\}$ represent the past history of the user u. where Q_i is the query and D_i is the concatenation of all the relevant documents for the query Q_i and let $D_i = \{w_1, w_2, ..., w_n\}$ be the words in it. The user profile learnt from the past history of user consists of the following triples of the form $(q, w_i, p(q|w_i))$ where q is a word in the query Q_i and w_i is a word in the document D_i.

Translation model is typically learnt from parallel texts i.e a set of translation pairs consisting of source and target language sentences. In learning the user profile, we first extract parallel texts from the past history of the user and then learn the translation model which is essentially the user profile. In the subsections below, we describe the process in detail.

5.1.1 Extracting Parallel Texts

By viewing documents as samples of a verbose language and the queries as samples of a concise language, we can treat each document-query pair as a translation pair, i.e. a pair of texts written in the verbose language and the concise language respectively. The extracted parallel texts consists of pairs of the form $\{Q_i, D_{rel}\}$ where D_{rel} is the concatenation of contexts extracted from all relevant document for the query Q_i.

We believe that short snippets extracted in the context of the query would be better candidates for D_{rel} than using the whole document. This is because there can be a lot of noisy terms which need not right in the context of the query. We believe a short snippet usually N (we considered 15) words to the left and right of the query words, similar to a short snippet displayed by search engines can better capture the context of the query. In deed we experimented with different context sizes for D_{rel}. The first is using the whole document i.e., considering the query and concatenation of all the relevant documents as a pair in the parallel texts extracted which is called $D_{documents}$ The second is using just a short text snippet from the document in the context of query instead of the whole document which is called $D_{snippets}$ Details are described in the experiments section.

5.1.2 Learning Translation Model

According to the standard statistical translation model (Brown et al., 1993), we can find the optimal model M^* by maximizing the probability of generating queries from documents or

$$M^* = \arg\max_M \prod_{i=1}^{N} P(Q_i|D_i, M)$$

qw	dw	P(qw|dw,u)
journal	kdd	0.0176
journal	conference	0.0123
journal	journal	0.0176
journal	sigkdd	0.0088
journal	discovery	0.0211
journal	mining	0.0017
journal	acm	0.0088
music	music	0.0375
music	purchase	0.0090
music	mp3	0.0090
music	listen	0.0180
music	mp3.com	0.0450
music	free	0.0008

Table 1: Sample user profile

To find the optimal word translation probabilities $P(qw|dw, M*)$, we can use the EM algorithm. The details of the algorithm can be found in the literature for statistical translation models, such as (Brown et al., 1993).

IBM Model1 (Brown et al., 1993) is a simplistic model which takes no account of the subtler aspects of language translation including the way word order tends to differ across languages. Similar to earlier work (Berger and Lafferty, 1999), we use IBM Model1 because we believe it is more suited for IR because the subtler aspects of language used for machine translation can be ignored for IR. GIZA++ (Och and Ney, 2003), an open source tool which implements the IBM Models which we have used in our work for computing the translation probabilities. A sample user profile learned is shown in Table 1.

5.2 Re-ranking

Re-ranking is a phase in personalized search where the set of documents matching the query retrieved by a general search engine are re-scored using the user profile and then re-ranked in descending order of rank of the document. We follow a similar approach in our work.

Let $\mathcal{D}$ be set of all the documents returned by the search engine. The rank of each document D returned for a query Q for user u is computing using his user profile as shown in Equation 1.

$$P(Q|D,u) = \prod_{q_i \in Q} \alpha P(q_i|GE) + (1-\alpha) \sum_{w \in D} P(q_i|w,u) P(w|D) \quad (1)$$

where $P(q_i|GE)$ is the smoothed or general probability obtained from a large general corpus. $P(q_i|w,u)$ is an entry in the translation model of the user. It represents the probability of generation of the query word q_i for a word w in the document. $P(w|D)$ is the probability of the word w in the document and α is a weighting parameter which lies between 0 and 1.

6 Experiments

We performed experiments evaluating our approach on data set consisting of 7 users. Each user submitted a number of queries to a search engine (Google). For each query, the user examined the top 10 documents and identified the set of relevant documents. Table 2 gives the statistics of the data sets. There is no repetition of query for any user though repetition of some words in the query exists (see Table 2). The document collection consists of top 20 documents from google which is actually the set of documents seen by the user while accessing the relevance of the documents. In all, the total size of the document collection was 3,469 documents. We did not include documents of type doc and pdf files.

To evaluate our approach, we use the 10-fold cross-validation strategy (Mitchell, 1997). We divide the data of each user into 10 sets each having (approximately) equal number of search queries (For example, for user1 had 37 queries in total, we divided this into 10 sets with 4 queries each approximately). Learning of user profile is done 10 times, each time leaving out one of the sets from training, but using only the omitted subset for testing. Performance is computed in the testing phase for each time and average of the 10 times is taken. In the testing phase, we take each query and re rank the results using the proposed approach using his profile learned from nine other sets. For measuring performance for each query, we compute Precision @10 (P@10), a widely used metric for evaluating personalized search algorithms. It is defined as the proportion of relevant documents among the top 10 results for the given ranking of documents. P@10 is computed by comparing with the relevant documents present in the data. All the values presented in the tables are average values which are averaged over all queries for each user, unless otherwise specified. We used Lucene[1], an open source search engine as the general search engine to first retrieve a

[1] http://lucene.apache.org

User	No. Q	% of Unique words in Q	Total Rel	Avg. Rel
1	37	89	236	6.378
2	50	68.42	178	3.56
3	61	82.63	298	4.885
4	26	86.95	101	3.884
5	33	80.76	134	4.06
6	29	78.08	98	3.379
7	29	88.31	115	3.965

Table 2: Statistics of the data set of 7 users

Set	Contextless	Proposed
User1	0.1433	0.1421
User2	0.1426	0.2445
User3	0.1016	0.1216
User4	0.0557	0.1541
User5	0.1877	0.3933
User6	0.1566	0.3941
User7	0.1	0.1833
Avg	0.1268	**0.2332**

Table 3: Precision @10 results for 7 users

set of results matching the query.

6.0.1 Comparison with Contextless Ranking

We test the effectiveness of our user profile by comparing with a contextless ranking algorithm. We used a generative language modeling for IR as the context less ranking algorithm (Query Likelihood model (Ponte and Croft, 1998; Song and Croft., 1999)). This is actually the simplest version of the model described in Equation 1. Each word w can be translated only as itself that is the translation probabilities (see Equation 1) are "diagonal".

$$P(q_i|w,u) = \begin{cases} 1 & \text{if } q = w \\ 0 & \text{Otherwise} \end{cases}$$

This serves as a good baseline for us to see how well the translation model actually captured the user information. For fair testing similar to our approach, for each query, we first retrieve results matching a query using a general search engine (Lucene). Then we rank the results using the formula shown in Equation 2.

$$P(Q|D) = \prod_{q_i \in Q} \alpha P(q_i|GE) + (1-\alpha) P(q_i|D) \quad (2)$$

We used IBM Model1 for learning the translation model (i.e., the user profile). The general English probabilities are computed from all the documents in the lucene's index. Similar to earlier works (Berger and Lafferty, 1999), we simply set the value of α to be 0.05. The values reported are P@10 values average over all 10 sets and the queries for the respective user. Table 3 clearly shows the improvement brought in by the user profile.

6.0.2 Experiments with Different Models

We performed an experiment to see if different training models for learning the user profile affected the performance. We experimented with two models. The first is a basic model and used in earlier work, IBM Model1. The second is using the GIZA++ default parameters. We observed that user profile learned using IBM Model1 outperformed that using GIZA++ default parameters. We believe this is because, IBM Model1 is more suited for IR because the subtler aspects of language used for machine translation (which are used in GIZA++ default parameters) can be ignored for IR. We obtained an average P@10 value of 0.2333 for IBM Model1 and 0.2075 for GIZA++.

Training Model	Document Test	Snippet Test
IBM Model1		
Document Train	0.2062	0.2028
Snippet Train	0.2333	**0.2488**
GIZA++		
Document Train	0.1799	0.1834
Snippet Train	0.2075	0.2034

Table 4: Summary of Comparison of different Models and Contexts for learning user profile

6.0.3 Snippet Vs Document

In extracting parallel texts consists of pairs of the form $\{Q_i, D_{rel}\}$ where D_{rel} is the concatenation of contexts extracted from all relevant document for the query Q_i we experimented with different context sizes for D_{rel}.

We believe that a short snippet extracted in the context of the query would be better candidate for D_{rel} than using the whole document. This is because there can be a lot of noisy terms which need not useful in the context of the query. We believe a short snippet usually N (we considered 15) words to the left and right of the query words, similar to a short snippet displayed by search engines can better

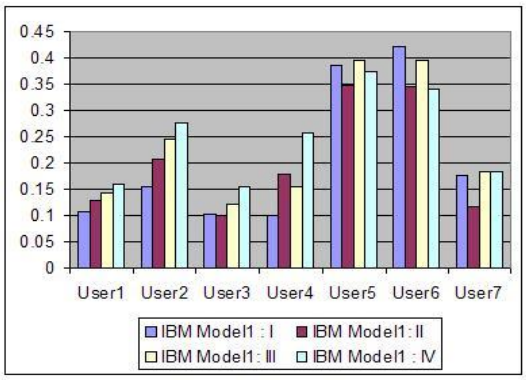


Figure 1: Comparison of Snippet Vs Document Training using IBM Model1 for training.
IBM Model1 : I - Document Training and Document Testing, IBM Model1 : II - Document Training and Snippet Testing, IBM Model1 : III - Snippet Training and Document Testing, IBM Model1 : IV - Snippet Training and Snippet Testing

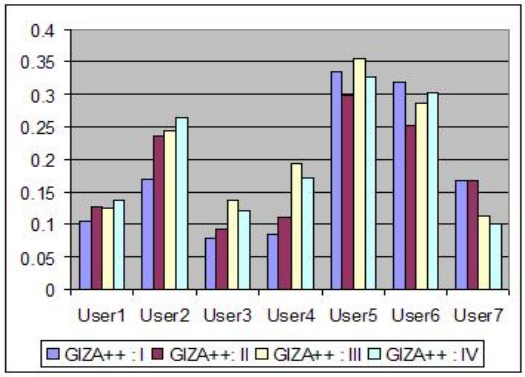


Figure 2: Comparison of Snippet Vs Document Training using GIZA++ Default parameters for training.
GIZA++:I - Document Training and Document Testing, GIZA++:II - Document Training and Snippet Testing, GIZA++:III - Snippet Training and Document Testing, GIZA++:IV - Snippet Training and Snippet Testing

capture the context of the query.

We experimented with two context sizes. The first is using the whole document i.e., considering the query and concatenation of all the relevant documents as a pair in the parallel texts extracted which is called $D_{documents}$. The second is using just a short text snippet from the document in the context of query instead of the whole document which is called $D_{snippets}$. The user profile learning from pairs of parallel texts $\{Q, D_{documents}\}$ is called *Document Train*. The user profile learning from pairs of parallel texts $\{Q, D_{snippets}\}$ is called *Snippet Train*. The user profiles are trained using both IBM Model1 and GIZA++ and comparison of the two is shown in Table 4.

We also experimented with the size of the context used for testing. Using the document for re-ranking as shown in Equation 1 (called *Document Test*) [2] and using just a short snippet extracted from the document for testing (called *Snippet Test*). Table 4 shows the average P@10 over the 10 sets and all queries and users.

We observed that, not only did the model used for training affected P@10, but also the data used in training and testing, whether it was a snippet or document, showed a large variation in the performance. Training using IBM Model1 using the snippet and testing using snippet achieved the best results. This is in agreement with the discussion that the snippet surrounding the query captures the context of the query better than a document which may contain many words that could possibly be unrelated to the query, therefore diluting the strength of the models learnt. The detailed results for all the users are shown in Figure 1 and Figure 2.

[2]It is to be noted that *Snippet Train* and *Document Test* and training using IBM Model1 is the default configuration used for all the reported results unless explicitly specified.

7 Conclusions and Future Work

Relevance feedback from the user has been used in various ways to improve the relevance of the results for the user. In this paper we have proposed a novel usage of relevance feedback to effectively model the process of query formulation and better characterize how a user relates his query to the document that he intends to retrieve. We applied a noisy channel model approach for the query and the documents in a retrieval process. The user profile was modeled using the relevance feedback obtained from the user as the probabilities of translation of query to document in this noisy channel. The user profile thus learnt was applied in a re-ranking phase to rescore the search results retrieved using general information retrieval models. We evaluate the usage of our approach by conducting experiments using relevance feedback data collected from users of a popular search engine. Our experiments have resulted in

some valuable observations that learning these user profiles using snippets surrounding the results for a query show better performance than when learning from entire documents. In this paper, we have only evaluated explicit relevance feedback gathered from a user and performed our experiments. As part of future work, we would like to evaluate our approach on implicit feedback gathered probably as click-through data in a search engine, or on the client side using customized browsers.

References

Adam Berger and John D. Lafferty. 1999. Information retrieval as statistical translation. In *Research and Development in Information Retrieval*, pages 222–229.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

W. Bruce Croft, Stephen Cronen-Townsend, and Victor Larvrenko. 2001. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*.

Jamie Allan et. al. 2003. Challenges in information retrieval language modeling. In *SIGIR Forum*, volume 37 Number 1.

K. Sugiyama K. Hatano and M. Yoshikawa. 2004. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of WWW 2004*, page 675 684.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *Research and Development in Information Retrieval*, pages 120–127.

F. Liu, C. Yu, and W. Meng. 2002. Personalized web search by mapping user queries to categories. In *Proceedings of the eleventh international conference on Information and knowledge management, ACM Press*, pages 558–565.

Tom Mitchell. 1997. *Machine Learning*. McGrawHill.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, pages 275–281.

A. Pretschner and S. Gauch. 1999. Ontology based personalized search. In *ICTAI.*, pages 391–398.

J. J. Rocchio. 1971. Relevance feedback in information retrieval, the smart retrieval system. *Experiments in Automatic Document Processing*, pages 313–323.

G. Salton and C. Buckley. 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society of Information Science*, 41:288–297.

Xuehua Shen, Bin Tan, and Chengxiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of CIKM 2005*.

F. Song and W. B. Croft. 1999. A general language model for information retrieval. In *Proceedings on the 22nd annual international ACM SIGIR conference*, page 279280.

Micro Speretta and Susan Gauch. 2004. Personalizing search based on user search histories. In *Thirteenth International Conference on Information and Knowledge Management (CIKM 2004)*.

Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of ACM SIGIR'01*, pages 334–342.

Repurposing Theoretical Linguistic Data for Tool Development and Search

Fei Xia
University of Washington
Seattle, WA 98195
fxia@u.washington.edu

William D. Lewis*
Microsoft Research
Redmond, WA 98052-6399
wilewis@microsoft.com

Abstract

For the majority of the world's languages, the number of linguistic resources (e.g., annotated corpora and parallel data) is very limited. Consequently, supervised methods, as well as many unsupervised methods, cannot be applied directly, leaving these languages largely untouched and unnoticed. In this paper, we describe the construction of a resource that taps the large body of linguistically analyzed language data that has made its way to the Web, and propose using this resource to bootstrap NLP tool development.

1 Introduction

Until fairly recently, most NLP research has focused on the ten or so majority languages of the world, the canonical *high density* languages. *Low density*, or *resource poor languages* (RPLs), have more recently captured the interest of NLP research, mostly because of recent advances in computational technologies and computing power. As indicated by their name, RPLs suffer from a lack of resources, namely data. Supervised learning techniques generally require large amounts of annotated data, something that is nonexistent or scare for most RPLs. A greater number of RPLs, however, have raw data that is available, and the amount and availability of this raw data is increasing every day as more of it makes its way to the Web. Likewise, advances in un- and semi-supervised learning techniques have made raw data more readily viable for tool development. Still, however, such techniques often require "seeds", or "prototypes" (c.f., (Haghighi and Klein, 2006)) which are used to prune search spaces or direct learners.

An important question is how to create such seeds for the hundreds to thousands of RPLs. We describe the construction of a resource that taps the large body of linguistically analyzed language data that has made its way to the Web, and propose using this resource as a means to bootstrap NLP tool development. Interlinear Glossed Text, or IGT, a semi-structured data type quite common to the field of linguistics, is used to present data and analysis for a language and is generally embedded in scholarly linguistic documents as part of a larger analysis. IGT's unique structure — effectively each instance consists of a bitext between English and some target language — can be easily enriched through alignment and projection (e.g., (Yarowsky and Ngai, 2001), (Hwa et al., 2002)). The reader will note that the IGT instance in Example (1) consists of a bitext between some target language on the first line, or the *target line* (in this case in Welsh), and a third line in English, the *translation line*. The canonical IGT form, which this example is representative of, has intervening linguistic annotations and glosses on a second line, the *gloss line*. Because the gloss line aligns with words and morphemes on the target line, and contains glosses that are similar to words on the translation line, it can serve as a bridge between the target and translation lines; high word alignment accuracy between the three lines can be achieved without requiring parallel data or bilingual dictionaries (Xia and Lewis, 2007). Furthermore, the gloss line provides additional information about the target language data, such as a variety of grammatical annotations, including verbal and tense markers (e.g., 3sg), case markers, etc., all of which can provide useful knowledge about the language.

(1) Rhoddodd yr athro lyfr i'r bachgen ddoe
gave-3sg the teacher book to-the boy yesterday
"The teacher gave a book to the boy yesterday"
(Bailyn, 2001)

ODIN, the Online Database of INterlinear text (Lewis, 2006), is a resource built over the past few years from data harvested from scholarly documents. Currently, ODIN has over 41,581 instances of IGT for 944 languages, and the number of IGT instances is expected to double or triple in the near-term as new methods for collecting data are brought online. Although the number of instances per language varies, e.g., the maximum currently is 2,891 instances (for

*The work described in this document was done while Lewis was faculty at the University of Washington.

Table 1: The numbers of languages in ODIN

Range of IGT instances	# of languages	# of instances	% of instances
1000-2891	10	15019	36.11
500-999	11	8111	19.50
250-499	18	6274	15.08
100-249	22	3303	7.94
50-99	38	2812	6.76
25-49	60	2089	5.02
10-24	127	1934	4.65
1-9	658	2039	4.91

Japanese), and the overall number per language may appear small, it is still possible to harvest significant value from IGT for targeted RPLs. In this paper, we present the ODIN database and methods used to create it. We also present methods we have employed to enrich IGT in order to make it more readily useful for bootstrapping NLP tools. Because the canon of knowledge embodied in the hundred or so years of linguistic analysis remains virtually untapped by the NLP community, we provide a bridge between the communities by providing linguistic data in a way that NLP researchers will find useful. Likewise, because IGT is a common linguistic data type, we provide a search facility over these data, which has already been found to be quite useful to the theoretical linguistics community.

2 Building ODIN

ODIN currently has 41,581 IGT instances for 944 languages. Table 1 shows the number of languages that fall into buckets defined by the number of IGT instances for each language. For instance, the fourth row ("bucket") says that 22 languages each have 100 to 249 IGT instances, and the 3,303 instances in this bucket account for 7.94% of all instances. ODIN is built in three steps, as described below.[1]

2.1 Crawling for IGT documents

Because a large number of instances of IGT exist on the Web,[2] we have focused on searching for these instances. The major difficulty with locating documents that contain IGT, however, is reducing the size of the search space. We decided very early in the development of ODIN that unconstrained Web crawling was too time and resource intensive a process to be feasible, mostly due to the Web's massive size. We discovered that highly focused metacrawls were far more fruitful. Metacrawling essentially involves throwing queries against an existing search engine, such as Google, Yahoo or MSN Live, and crawling only the pages returned by those queries. We found that the most successful queries were those that used strings contained within IGT itself, e.g. grammatical annotations, or *grams*, such as 3sg, NOM, ACC, etc. In addition, we found precision increased when we included two or more search terms per query, with the most successful queries being those which combined grams and language names. Thus, for example, although NOM alone returned a large number of linguistic documents, NOM combined with ACC (or any other high frequency term), or a language name, returned a far less noisy and far more relevant set of documents.

Other queries we have developed include: queries by language names and language codes (drawn from the Ethnologue database (Gordon, 2005), which contains about 40,000 language names and their variants), by linguists' names and the languages they work on (drawn from the Linguist List's linguist database), by linguistically relevant terms (drawn from the SIL linguistic glossary), and by particular words or morphemes found in IGT and their grammatical markup. Table 2 shows the statistics for the most successful crawls and their related search term "types". Calculated from the top 100 queries for each type, the table presents the most successful query types, the average number of documents returned for each, the average number of documents in which IGT was actually found, and the average number of IGT instances netted by each query. The most relevant measure of success is the number of IGT instances returned (the obvious focus of our crawling); in turn, the most successful query types are those which contain a combination of grams and language names.[3]

[1] The work of creating ODIN, in some ways, speaks to the need of standardizing IGT (perhaps along with other linguistic data types) such that both humans and machines can more readily consume the data. Some recent efforts to develop standards for encoding IGT (e.g., (Hughes et al., 2003), (Bickel et al., 2004)) have met with limited success, however, since they have not been widely recognized and even less frequently adopted. Over time it is our hope that these or other standards will see wider use thus eliminating the need for much of the work proposed here.

[2] Although we have no direct data about the total number of IGT instances that exist on the Web, we hypothesize that the total supply is at least several hundred thousand instances. Given that ODIN contains 41,581 instances which have been extracted from approximately 3,000 documents, and given that we have located at least 60,000 more documents that might contain IGT, we feel our estimate to be reasonable.

[3] Note that target documents are often returned by multiple queries. For instance, the documents returned by "NOM+ACC+Icelandic" will also be returned by the individual query terms "NOM", "ACC", and "Icelandic".

Table 2: The Most Successful Query Types

Query **Type**	Avg # docs	Avg # docs w/ IGT	Avg # IGTs
Gram(s)	1184	239	50
Language name(s)	1314	259	33
Both grams and names	1536	289	77
Language words	1159	193	0

2.2 IGT detection

After crawling, the next step is to identify IGT instances in the retrieved documents. This is a difficult task for which machine learning methods are well suited.

2.2.1 Difficulty in IGT detection

The canonical form of IGT, as presented in Section 1, consists of three parts and each part is on a single line. However, many IGT instances do not follow the canonical format for several reasons. First, when IGT examples appear in a group, very often the translation or glosses are dropped for some examples in the group because the missing parts can be recovered from the context, resulting in *two-part* IGT. In other cases, some IGT examples include multiple target transcriptions (e.g., one part in the native script, and another in a latin transliteration) or even, in rare cases, multiple translations.

Second, dictated by formatting constraints, long IGT examples may need to be wrapped one or more times, and there are no conventions on how wrapping should be done, nor how many times it can be done. For short IGT examples, sometimes linguists put the translation to the right of the target line rather than below it. As a result, each part of IGT examples may appear on multiple lines and multiple parts can appear on a single line.

Third, most IGT-bearing documents on the Web are in PDF, and the PDF-to-text conversion tools will sometimes corrupt IGT instances (most often on the target line). In some instances, some words or morphemes on the target line are inadvertently dropped in the conversion, or are displaced up or down a line. Finally, an IGT instance could fall into multiple categories. For instance, a two-part IGT instance could have a corrupted target line. All of this makes the detection task difficult.

2.2.2 Applying machine learning methods

The first system that we designed for IGT detection used regular expression "templates", effectively looking for text that resembled IGT. An example is shown in (2), which matches any three-line instance (e.g., the IGT instance in (1)) such that the first line starts with an example number (e.g., *(1)*) and the third line starts with a quotation mark.

(2)
```
\s*\(\d+\).*\n
\s*.*\n
\s*\[‘’"].*\n
```

Unfortunately, this approach tends to over-select when applied to the documents crawled from the Web. Further, many true IGT instances do not match any of hand-written templates due to the issues mentioned in the previous section. As a result, both precision and recall are quite low (see Table 4).

Given the irregular structure of IGT instances, a statistical system is likely to outperform a rule-based system. In our second system, we treat the IGT detection task as a sequence labeling problem, and apply machine learning methods to the task: first, we train a learner and use it to tag each line in a document with a tag in a pre-defined tag set; then we convert the best tag sequence into a span sequence. A span is a (start, end) pair, which indicates the beginning and ending line numbers of an IGT instance.

Among all the tagging schemes we experimented with (including the standard BIO tagging scheme), the following 5-tag scheme works the best on the development set: The five tags are BL (any blank line), O (outside IGT that is not a BL), B (the first line in an IGT), E (the last line in an IGT), I (inside an IGT that is not a B, E, or BL).

For machine learning, we use four types of features:

F_1: The words that appear on the current line. These are the features typically used in a text classification task.

F_2: Sixteen features that look at various cues for the presence of an IGT. For example, whether the line starts with a quotation, whether the line starts with an example number (e.g., (1)), and whether the line contains a large portion of hyphenated or non-English tokens.

F_3: In order to find good tag sequences, we include features for the tags of the previous two lines.

F_4: The same features as in F_2, but they are checked against the neighboring lines. For instance, if a feature f_5 in F_2 checks whether the current line contains a citation, f_5^{+1} checks whether the next line contains a citation.

After the lines in a document are tagged by the learner, we identify IGT instances by finding all the spans in the document that match the "B [I | BL]* E" pattern; that is, the span starts with a B, ends with an E, and has zero or more I or BL in between.[4]

[4]Other heuristics for converting tag sequences to span sequences produce similar results.

Table 3: Data sets for the IGT detection experiments

	# files	# lines	# IGTs
Training data	41	39127	1573
Dev data	10	8932	447
Test data	10	14592	843

2.2.3 Experimental results

To evaluate the two detectors, we randomly selected 61 ODIN documents and manually marked the occurrence of IGT instances. The files were then split into training, development, and test sets, and the size of each set is shown in Table 3. The annotation speed was about four thousand lines per hour. Each file in the development and test sets was annotated independently by two annotators, and the inter-annotator agreement (f-score) on IGT boundary was 93.74% when using *exact match* (i.e., two spans match *iff* they are identical). When *partial match* (i.e., two spans match *iff* they overlap) was used, the f-score increased to 98.66%.

We used four machine learning algorithms implemented in Mallet (McCallum, 2002): decision tree, Naive Bayes, maximum entropy (MaxEnt), and conditional random field (CRF).[5] Table 4 shows the MaxEnt model's performance on the development set with different combinations of features: the highest f-score for exact match in each group is marked in boldface.[6] In addition to exact and partial match results, we also list the number of spans produced by the system (cf. the span number in the gold standard is 447) and the classification accuracy (i.e., the percent of lines receiving correct labels). The results for CRF are very similar to those for MaxEnt, and both outperform decision tree and Naive Bayes.

Several observations are in order. First, as expected, the machine learning approach outperforms the regular expression approach. Second, although F_2 contains only sixteen features, it works much better than F_1, which uses all the words occurring in the training data. Third, F_4 works much better than F_3 in capturing contextual information, mainly because F_4 allows the learner to take into account the information that appears on both the preceding lines and the succeeding lines.[7] Last, adding F_1 and F_3 to the $F_2 + F_4$ system offers a modest but statistically significant gain.

Table 5 shows the results on the test data. The performance of MaxEnt on this data set is slightly worse than on the development set mainly because the test set contains much more corrupted data (due to pdf-to-text conversion) than both the training and development sets.[8] Nevertheless, the machine learning approach outperforms the regex approach significantly, reducing the error rate by 52.3%. In addition, the partial match results are much better than exact match results, indicating that many span errors could be potentially fixed by postprocessing.

2.3 Manual review and language ID

About 45% of IGT instances in the current ODIN database were manually checked to verify IGT boundaries and to identify the language names of the target lines. Subsequently, we trained several language ID algorithms with the labeled data, and used them to label the remaining 55% of the IGT instances in ODIN automatically.

The language ID task in this context is different from a typical language ID task in several ways. First, the number of languages in IGT is close to a thousand or even more. In contrast, the amount of training data for many of the languages is very limited; for instance, hundreds of languages have less than 10 sentences, as shown in Table 1. Second, some languages in the test data might never occur in the training data, a problem that we shall call the *unknown language problem*. Third, the target sentences in IGT are very short (e.g., a few words), making the task more challenging. Fourth, for languages that do not use a latin-based writing system, the target sentences are often transliterated, making the character encoding scheme less informative. Last, the context, such as the language names occurring in the document, provides important cues for the language ID of IGT instances.

Given these properties, applying common language ID algorithms directly will not produce satisfactory results. For instance, Cavnar and Trenkle's N-gram-based algorithm yields an accuracy of as high as 99.8% when tested on newsgroup articles in eight languages (Cavnar and Trenkle, 1994).[9]

[5] For the first three methods, we implemented beam search to find the best tag sequences; and for CRF, we used features in F1, F2, and F4, as the model itself incorporates the information about previous tags already.

[6] F_4 is an extension of F_2, so every combination with F_4 should include F_2 as well. Also, F_3 should not be used alone. Therefore, Table 4 in fact lists all the possible feature combinations.

[7] The window for F_4 is set empirically to [-2,3]; that is, F_4 uses the information from the preceding two lines and the succeeding three lines.

[8] The corruption not only affects the target lines, but also the layout of IGT (e.g., the indentation of the three lines). As a result, features in F_2 and F_4 are not as effective as for the development set. Since the regex template approach uses fewer layout features, its performance is not affected as much.

[9] The accuracy ranges from 92.9% to 99.8% depending on the article length and a model parameter called *profile*

Table 4: Performance on the development set (the span number in the gold standard is 447)

Features	System span num	Classification accuracy	Exact match			Partial match		
			prec	recall	**fscore**	prec	recall	fscore
Regex templates	269	N/A	68.40	41.16	**51.40**	99.26	59.73	74.58
F_1	130	81.50	68.46	19.91	30.85	97.69	28.41	44.02
F_2	405	93.28	58.27	52.80	**55.40**	95.56	86.58	90.85
$F_1 + F_3$	180	80.26	61.67	24.83	35.40	81.11	32.66	46.57
$F_1 + F_2$	420	94.42	63.09	59.28	61.13	93.81	88.14	90.88
$F_2 + F_3$	339	92.68	75.81	57.49	65.39	93.21	70.69	80.40
$F_2 + F_4$	456	96.91	80.92	82.55	**81.73**	93.64	95.53	94.57
$F_1 + F_2 + F_3$	370	93.39	75.14	62.20	68.05	93.51	77.40	84.70
$F_1 + F_2 + F_4$	444	97.00	84.68	84.11	84.40	95.95	95.30	95.62
$F_2 + F_3 + F_4$	431	97.79	86.77	83.67	**85.19**	97.68	94.18	95.90
$F_1 + F_2 + F_3 + F_4$	431	98.00	90.02	86.80	**88.38**	97.22	93.74	95.44

Table 5: Performance on the test set (the span number in the gold standard is 843)

Features	System span num	Classification accuracy	Exact match			Partial match		
			prec	recall	**fscore**	prec	recall	fscore
Regex templates	587	N/A	74.95	52.19	**61.54**	98.64	68.68	80.98
F_2	719	92.45	57.02	48.64	52.50	94.02	80.19	86.56
$F_2 + F_4$	849	95.66	75.50	76.04	75.77	93.76	94.42	94.09
$F_2 + F_3 + F_4$	831	95.95	77.14	76.04	76.58	95.19	93.83	94.50
$F_1 + F_2 + F_3 + F_4$	830	96.83	82.29	81.02	**81.65**	96.51	95.02	95.76

However, when we ran the same algorithm on the IGT data, the accuracy was only 50.2%.[10] In contrast, a heuristic approach that predicts the language ID according to the language names occurring in the document yields an accuracy of 65.6%.

Because the language name associated with an IGT instance almost always appears somewhere in the document, we propose to treat the language ID task as a reference resolution problem, where IGT instances are the *mentions* and the language names appearing in the document are the *entities*. A language identifier simply needs to link the mentions to the entities, allowing us to apply any good resolution algorithms such as (Soon et al., 2001; Ng, 2005; Luo, 2007) and to provide an elegant solution to the unknown language problem. More detail on this approach will be reported elsewhere.

3 Using ODIN

We see ODIN being used in a number of different ways. In another study (Lewis and Xia, 2008), we demonstrated a method for using ODIN to discover interesting and computationally relevant typological features for hundreds of the world's languages automatically. In this section we present two more uses for ODIN's data: bootstrapping NLP tools (specifically taggers), and providing search over ODIN's data (as a kind of large-scale multi-lingual search).

3.1 IGT for bootstrapping NLP tools

Since the target line in IGT data does not come with annotations (e.g., POS tags), it is first necessary to enrich it. Once enriched, the data can be used as a bootstrap for tools such as taggers.

3.1.1 Enriching IGT

In a previous study (Xia and Lewis, 2007), we proposed a three-step process to enrich IGT data: (1) parse the English translation with an English parser and convert English phrase structures (PS) into dependency structures (DS) with a head percolation table (Magerman, 1995), (2) align the target line and the English translation using the gloss line, and (3) project the syntactic structures (both PS and DS) from English onto the target line. For instance, given the IGT example in Ex (1), the enrichment algorithm will produce the word alignment in Figure 1 and the syntactic structures in Figure 2.

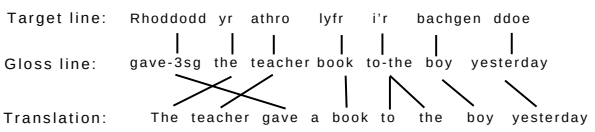


Figure 1: Aligning the target line and the English translation with the help of the gloss line

length.

[10] The setting for our preliminary experiments is as follows: there are 10,415 IGT instances over 549 languages in the training data, and 3064 instances in the test data. The language names of about 12.2% of IGT instances in the test data never appear in the training data.

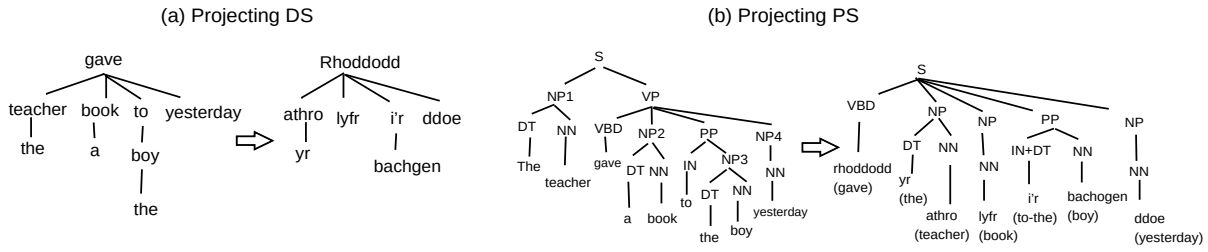


Figure 2: Projecting syntactic structure from English to the target language

We evaluated the algorithm on a small set of 538 IGT instances for several languages. On average, the accuracy of the English DS (i.e., the percentage of correct dependency links in the DS) is 93.48%; the f-score of the word alignment links between the translation and target lines is 94.03%, and the accuracy of the target DS produced by the projection algorithm is 81.45%. When we replace the automatically generated English DS and word alignment with the ones in the gold standard, the accuracy of target DS increases significantly, from 81.45% to 90.64%. The details on the algorithms and the experiments can be found in (Xia and Lewis, 2007).

3.1.2 Bootstrapping NLP tools

The enriched data produced by the projection algorithms contains (1) the English DS and PS produced by an English parser, (2) the word alignment among the three parts of IGT data, and (3) the target DS and PS produced by the projection algorithm. From the enriched data, various kinds of information can be extracted. For instance, the target syntactic structures form small monolingual treebanks, from which grammars in various formalisms can be extracted (e.g., (Charniak, 1996)). The English and target syntactic structures form parallel treebanks, from which transfer rules and translation lexicon can be extracted and used for machine translation (e.g., (Meyers et al., 2000; Menezes, 2002; Xia and McCord, 2004)).

There are many ways of using the enriched data to bootstrap NLP tools. Suppose we want to build a POS tagger. Previous studies on unsupervised POS tagging can be divided into several categories according to the kind of information available to the learner. The first category (e.g., (Kupiec, 1992; Merialdo, 1994; Banko and Moore, 2004; Wang and Schuurmans, 2005)) assumes there is a lexicon that lists the allowable tags for each word in the text. The common approach is to use the lexicon to initialize the emission probability in a Hidden Markov Model (HMM), and run the Baum-Welch algorithm (Baum et al., 1970) on a large amount of unlabeled data to re-estimate transition and emission probability. The second category uses unlabeled data only (e.g., (Schütze, 1995; Clark, 2003; Biemann, 2006; Dasgupta and Ng, 2007)). The idea is to cluster words based on morphological and/or distributional cues. Haghighi and Klein (2006) showed that adding a small set of prototypes to the unlabeled data can improve tagging accuracy significantly.

The tagged target lines in the enriched IGT data can be incorporated in each category of work mentioned above. For instance, the frequency collected from the data can be used to bias initial transition and emission probabilities in an HMM model; the tagged words in IGT can be used to label the resulting clusters produced by the word clustering approach; the frequent and unambiguous words in the target lines can serve as prototype examples in the prototype-driven approach (Haghighi and Klein, 2006). Finally, we can apply semi-supervised learning algorithms (e.g., self-training (Yarowsky, 1995), co-training (Blum and Mitchell, 1998), and transductive support vector machines (Vapnik, 1998)), using the tagged sentences as seeds.

3.2 Search

One focus of ODIN is and has always been search: how can linguists find the data that they are interested in and how can the data be encoded in such a way as to accommodate the variety of queries that a linguist might ask. We currently allow four types of search queries: search by language name and code, search by language family, search by concept/gram, and search by linguistic constructions. The first allows the user to specify a language name or ISO code to search for, and allows the user to view documents that contain instances of IGT in that language, as well as the instances themselves. The second allows the user to specify a language family (families as specified in the Ethnologue), and returns similar results, except grouped by language. The third allows the user to select from a list of known grams, all of which have been mapped to a conceptual space

used by linguists (the GOLD ontology, (Farrar and Langendoen, 2003)).[11]

The final query type, the Construction Search is the most powerful and most innovative of the query facilities currently provided by ODIN. Rather than limiting search to just the content and markup natively contained within IGT, Construction Search searches over *enriched* content. For instance, a search for relative clauses can look for either the POS tag sequences that contain a noun followed by an appropriate relativizer, or the parse trees that contain an NP node with an NP child and a clause child. Currently, 15 construction queries have been implemented, with some 40 additional queries being evaluated and built. Note that currently construction queries are performed on the English translation, not on the target language data. As syntactic projection becomes more reliable, we will allow construction queries on the target language data and even queries on both the English and the target (e.g., for comparative linguistic analyses). For example, a query could be something like *Find examples where the target line uses imperfective aspect and is in active voice and the English translation uses passive voice.*

4 Conclusion and Future Directions

In this paper, we introduce Interlinear Glossed Text (IGT), a data type that has been rarely tapped by the NLP community, and describe the process of creating ODIN, a database of IGT data. We show that using machine learning methods can significantly improve the performance of IGT detection. We then demonstrate how IGT instances can be enriched and discuss several ways of using enriched data to bootstrap NLP tools such as POS taggers. Finally, we review the four types of linguistic search that are currently implemented in ODIN. All of the above show the value of ODIN as a resource for both NLP researchers and linguists. In the future, we plan to improve the IGT detection and language ID algorithms and will apply them to all the crawled documents. We expect the size of ODIN to grow dramatically. We also plan to use the enriched data to bootstrap taggers and parsers, starting with the ideas outlined in Section 3.1.2.

Acknowledgements This work has been supported, in part, by the Royalty Research Fund at the University of Washington. We would also like to thank Dan Jinguji for providing the preliminary results on language ID expriments, and three anonymous reviewers for their valuable comments.

[11] Most gram-to-concept mapping has been done by hand. We are currently exploring methods to use machine learning to enhance our ability to identify and map additional unknown grams (to be discussed elsewhere).

References

John Frederick Bailyn. 2001. Inversion, dislocation and optionality in Russian. In Gerhild Zybatow, editor, *Current Issues in Formal Slavic Linguistics.*

Michele Banko and Robert C. Moore. 2004. Part of Speech Tagging in Context. In *Proc. of the 20th International Conference on Computational Linguistics (Coling 2004)*, pages 556–561, Geneva, Switzerland.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statistics*, 41(1):164–171.

Balthasar Bickel, Bernard Comrie, and Martin Haspelmath. 2004. The Leipzig Glossing Rules: Conventions for interlinear morpheme-by-morpheme glosses (revised version). Technical report, Max Planck Institute for Evolutionary Anthropology and the Department of Linguistics of the University of Leipzig.

Chris Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*, pages 7–12, Sydney, Australia, July.

Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-training. In *Proc. of the Workshop on Computational Learning Theory (COLT-1998).*

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US.

Eugene Charniak. 1996. Treebank Grammars. In *Proc. of the 13th National Conference on Artificial Intelligence (AAAI-1996).*

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proc. of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2003).*

Sajib Dasgupta and Vincent Ng. 2007. Unsupervised part-of-speech acquisition for resource-scarce languages. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language*

Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 218–227.

Scott Farrar and D. Terence Langendoen. 2003. A linguistic ontology for the Semantic Web. *GLOT International*, 7(3):97–100.

Raymond G. Gordon, editor. 2005. *Ethnologue: Languages of the World*. SIL International, Dallas, TX, fifteenth edition.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the Human Language Technology Conference of the NAACL (HLT/NAACL 2006)*, pages 320–327, New York City, USA.

Baden Hughes, Steven Bird, and Cathy Bow. 2003. Interlinear text facilities. In *E-MELD 2003*, Michigan State University.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia, Pennsylvania.

J. Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6.

William Lewis and Fei Xia. 2008. Automatically Identifying Computationally Relevant Typological Features. In *Proc. of the Third International Joint Conference on Natural Language Processing (IJCNLP-2008)*, Hyderabad, India.

William Lewis. 2006. ODIN: A Model for Adapting and Enriching Legacy Infrastructure. In *Proc. of the e-Humanities Workshop, held in cooperation with e-Science 2006: 2nd IEEE International Conference on e-Science and Grid Computing*, Amsterdam.

Xiaoqiang Luo. 2007. Coreference or not: A twin model for coreference resolution. In *Proc. of the Conference on Human Language Technologies (HLT/NAACL 2007)*, pages 73–80, Rochester, New York.

David M. Magerman. 1995. Statistical Decision-Tree Models for Parsing. In *Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-1995)*, Cambridge, Massachusetts, USA.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Arul Menezes. 2002. Better contextual translation using machine learning. In *Proc. of the 5th conference of the Association for Machine Translation in the Americas (AMTA 2002)*.

Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2).

Adam Meyers, Michiko Kosaka, and Ralph Grishman. 2000. Chart-based transfer rule application in machine translation. In *Proc. of the 18th International Conference on Computational Linguistics (COLING 2000)*.

Vincent Ng. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 157–164, Ann Arbor, Michigan.

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Proc. of the EACL*, pages 141–148.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4).

V. Vapnik. 1998. *Statistical learning theory*. Wiley-Interscience.

Qin Iris Wang and Dale Schuurmans. 2005. Improved Estimation for Unsupervised Part-of-Speech Tagging. In *Proc. of IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE 2005)*.

Fei Xia and William Lewis. 2007. Multilingual structural projection across interlinear text. In *Proc. of the Conference on Human Language Technologies (HLT/NAACL 2007)*, pages 452–459, Rochester, New York.

Fei Xia and Michael McCord. 2004. Improving a Statistical MT System with Automatically Learned Rewrite Patterns. In *Proc. of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

David Yarowsky and Grace Ngai. 2001. Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In *Proc. of the 2001 Meeting of the North American chapter of the Association for Computational Linguistics (NAACL-2001)*, pages 200–207.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL-1995)*, pages 189–196, Cambridge, Massachussets.

Computing Paraphrasability of Syntactic Variants Using Web Snippets

Atsushi Fujita **Satoshi Sato**
Graduate School of Engineering, Nagoya University
{fujita,ssato}@nuee.nagoya-u.ac.jp

Abstract

In a broad range of natural language processing tasks, large-scale knowledge-base of paraphrases is anticipated to improve their performance. The key issue in creating such a resource is to establish a practical method of computing semantic equivalence and syntactic substitutability, i.e., paraphrasability, between given pair of expressions. This paper addresses the issues of computing paraphrasability, focusing on syntactic variants of predicate phrases. Our model estimates paraphrasability based on traditional distributional similarity measures, where the Web snippets are used to overcome the data sparseness problem in handling predicate phrases. Several feature sets are evaluated through empirical experiments.

1 Introduction

One of the common characteristics of human languages is that the same concept can be expressed by various linguistic expressions. Such linguistic variations are called paraphrases. Handling paraphrases is one of the key issues in a broad range of natural language processing (NLP) tasks. In information retrieval, information extraction, and question answering, technology of recognizing if or not the given pair of expressions are paraphrases is desired to gain a higher coverage. On the other hand, a system which generates paraphrases for given expressions is useful for text-transcoding tasks, such as machine translation and summarization, as well as beneficial to human, for instance, in text-to-speech, text simplification, and writing assistance.

Paraphrase phenomena can roughly be divided into two groups according to their compositionality. Examples in (1) exhibit a degree of compositionality, while each example in (2) is composed of totally different lexical items.

(1) a. be in our favor $\Leftrightarrow$ be favorable for us
b. show a sharp decrease $\Leftrightarrow$ decrease sharply
(Fujita et al., 2007)

(2) a. burst into tears $\Leftrightarrow$ cried
b. comfort $\Leftrightarrow$ console
(Barzilay and McKeown, 2001)

A number of studies have been carried out on both compositional (morpho-syntactic) and non-compositional (lexical and idiomatic) paraphrases (see Section 2). In most research, paraphrases have been represented with the similar templates, such as shown in (3) and (4).

(3) a. $N_1\ V\ N_2 \Leftrightarrow N_1$'s V-*ing* of N_2
b. $N_1\ V\ N_2 \Leftrightarrow N_2$ be V-*en* by N_1
(Harris, 1957)

(4) a. X wrote $Y \Leftrightarrow X$ is the author of Y
b. X solves $Y \Leftrightarrow X$ deals with Y
(Lin and Pantel, 2001)

The weakness of these templates is that they should be applied only in some contexts. In other words, the lack of applicability conditions for slot fillers may lead incorrect paraphrases. One way to specify the applicability condition is to enumerate correct slot fillers. For example, Pantel et al. (2007) have harvested instances for the given paraphrase templates based on the co-occurrence statistics of slot fillers and lexicalized part of templates (e.g. "deal with" in (4b)). Yet, there is no method which assesses semantic equivalence and syntactic substitutability of resultant pairs of expressions.

In this paper, we propose a method of directly computing semantic equivalence and syntactic substitutability, i.e., paraphrasability, particularly focusing on automatically generated compositional paraphrases (henceforth, syntactic variants) of predicate phrases. While previous studies have mainly targeted at words or canned phrases, we treat predicate phrases having a bit more complex structures.

This paper addresses two issues in handling phrases. The first is feature engineering. Generally speaking, phrases appear less frequently than single words. This implies that we can obtain only a small amount of information about phrases. To overcome the data sparseness problem, we investigate if the Web snippet can be used as a dense corpus for given phrases. The second is the measurement of paraphrasability. We assess how well the traditional distributional similarity measures approximate the paraphrasability of predicate phrases.

2 Related work

2.1 Representation of paraphrases

Several types of compositional paraphrases, such as passivization and nominalization, have been represented with some grammar formalisms, such as transformational generative grammar (Harris, 1957) and synchronous tree adjoining grammar (Dras, 1999). These grammars, however, lack the information of applicability conditions.

Word association within phrases has been an attractive topic. Meaning-Text Theory (MTT) is a framework which takes into account several types of lexical dependencies in handling paraphrases (Mel'čuk and Polguère, 1987). A bottleneck of MTT is that a huge amount of lexical knowledge is required to represent various relationships between lexical items. Jacquemin (1999) has represented the syntagmatic and paradigmatic correspondences between paraphrases with context-free transformation rules and morphological and/or semantic relations between lexical items, targeting at syntactic variants of technical terms that are typically noun phrases consisting of more than one word. We have proposed a framework of generating syntactic variants of predicate phrases (Fujita et al., 2007). Following the previous work, we have been developing three sorts of resources for Japanese.

2.2 Acquiring paraphrase rules

Since the late 1990's, the task of automatic acquisition of paraphrase rules has drawn the attention of an increasing number of researchers. Although most of the proposed methods do not explicitly eliminate compositional paraphrases, their output tends to be non-compositional paraphrase.

Previous approaches to this task are two-fold. The first group espouses the distributional hypothesis (Harris, 1968). Among a number of models based on this hypothesis, two algorithms are referred to as the state-of-the-art. DIRT (Lin and Pantel, 2001) collects paraphrase rules consisting of a pair of paths between two nominal slots based on point-wise mutual information. TEASE (Szpektor et al., 2004) discovers binary relation templates from the Web based on sets of representative entities for given binary relation templates. These systems often output directional rules such as exemplified in (5).

(5) a. X is charged by Y
$\Rightarrow$ Y announced the arrest of X
b. X prevent Y $\Rightarrow$ X lower the risk of Y

They are actually called inference/entailment rules, and paraphrase is defined as bidirectional inference/entailment relation[1]. While the similarity score in DIRT is symmetric for given pair of paths, the algorithm of TEASE considers the direction.

The other utilizes a sort of parallel texts, such as multiple translation of the same text (Barzilay and McKeown, 2001; Pang et al., 2003), corresponding articles from multiple news sources (Barzilay and Lee, 2003; Dolan et al., 2004), and bilingual corpus (Wu and Zhou, 2003; Bannard and Callison-Burch, 2005). This approach is, however, limited by the difficulty of obtaining parallel/comparable corpora.

2.3 Acquiring paraphrase instances

As reviewed in Section 1, paraphrase rules generate incorrect paraphrases, because their applicability conditions are not specified. To avoid the drawback, several linguistic clues, such as fine-grained classification of named entities and coordinated sentences, have been utilized (Sekine, 2005; Torisawa, 2006). Although these clues restrict phenomena to those appearing in particular domain or those describing coordinated events, they have enabled us to collect

[1] See http://nlp.cs.nyu.edu/WTEP/

paraphrases accurately. The notion of Inferential Selectional Preference (ISP) has been introduced by Pantel et al. (2007). ISP can capture more general phenomena than above two; however, it lacks abilities to distinguish antonym relations.

2.4 Computing semantic equivalence

Semantic equivalence between given pair of expressions has so far been estimated under the distributional hypothesis (Harris, 1968). Geffet and Dagan (2005) have extended it to the distributional inclusion hypothesis for recognizing the direction of lexical entailment. Weeds et al. (2005), on the other hand, have pointed out the limitations of lexical similarity and syntactic transformation, and have proposed to directly compute the distributional similarity of pair of sub-parses based on the distributions of their modifiers and parents. We think it is worth examining if the Web can be used as the source for extracting features of phrases.

3 Computing paraphrasability between predicate phrases using Web snippets

We define the concept of paraphrasability as follows:

> A grammatical phrase s is paraphrasable with another phrase t, iff t satisfies the following three:
>
> - t is grammatical
> - t holds if s holds
> - t is substitutable for s in some context

Most previous studies on acquiring paraphrase rules have evaluated resultant pairs from only the second viewpoint, i.e., semantic equivalence. Additionally, we assume that one of a pair (t) of syntactic variants is automatically generated from the other (s). Thus, grammaticality of t should also be assessed. We also take into account the syntactic substitutability, because head-words of syntactic variants sometimes have different syntactic categories.

Given a pair of predicate phrases, we compute their paraphrasability in the following procedure:

Step 1. Retrieve Web snippets for each phrase.

Step 2. Extract features for each phrase.

Step 3. Compute their paraphrasability as distributional similarity between their features.

The rest of this section elaborates on each step in turn, taking Japanese as the target language.

3.1 Retrieving Web snippets

In general, phrases appear less frequently than single words. This raises a crucial problem in computing paraphrasability of phrases, i.e., the sparseness of features for given phrases. One possible way to overcome the problem is to take back-off statistics assuming the independence between constituent words (Torisawa, 2006; Pantel et al., 2007). This approach, however, has a risk of involving noises due to ambiguity of words.

We take another approach, which utilizes the Web as a source of examples instead of a limited size of corpus. For each of the source and target phrases, we retrieve snippets via the Yahoo API[2]. The number of snippets is set to 500.

3.2 Extracting features

The second step extracts the features for each phrase from Web snippets. We have some options for feature set, feature weighting, and snippet collection.

Feature sets

To assess a given pair of phrases against the definition of paraphrasability, the following three sets of features are examined.

HITS: A phrase must appear in the Web if it is grammatical. The more frequently a phrase appears, the more likely it is grammatical.

BOW: A pair of phrases are likely to be semantically similar, if the distributions of words surrounding the phrases are similar.

MOD: A pair of phrases are likely to be substitutable with each other, if they share a number of instances of modifiers and modifiees.

To extract BOW features from sentences including the given phrase within Web snippets, a morphological analyzer MeCab[3] was firstly used; however, it resulted wrong POS tags for unknown words, and hurt statistics. Thus, finally ChaSen[4] is used.

To collect MOD features, a dependency parser CaboCha[5] is used. Figure 1 depicts an example of extracting MOD features from a sentence within Web snippet. A feature is generated from a *bunsetsu*, the Japanese base-chunk, which is either mod-

[2] http://developer.yahoo.co.jp/search/
[3] http://mecab.sourceforge.net/
[4] http://chasen.naist.jp/hiki/ChaSen/
[5] http://chasen.org/~taku/software/cabocha/

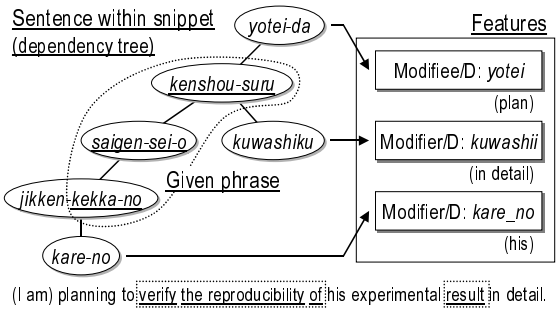


Figure 1: An example of MOD feature extraction. An oval in the dependency tree denotes a *bunsetsu*.

ifier or modifiee of the given phrase. Each feature is composed of three or more elements: (i) modifier or modifiee, (ii) dependency relation types (direct dependency, appositive, or parallel, c.f., RASP and MINIPAR), (iii) base form of the head-word, and (iv) case marker following noun, auxiliary verb and verbal suffixes if they appear. The last feature is employed to distinguish the subtle difference of meaning of predicate phrases, such as voice, tense, aspect, and modality. While Lin and Pantel (2001) have calculated similarities of paths based on slot fillers of subject and object slots, MOD targets at sub-trees and utilizes any modifiers and modifiees.

Feature weighting

Geffet and Dagan (2004) have reported on that the better quality of feature vector (weighting function) leads better results. So far, several weighting functions have been proposed, such as point-wise mutual information (Lin and Pantel, 2001) and Relative Feature Focus (Geffet and Dagan, 2004). While these functions compute weights using a small corpus for merely re-ranking samples, we are developing a measure that assesses the paraphrasability of arbitrary pair of phrases, where a more robust weighting function is necessary. Therefore we directly use frequencies of features within Web snippets as weight. Normalization will be done when the paraphrasability is computed (Section 3.3).

Source-focused feature extraction

Independent collection of Web snippets for each phrase of a given pair might yield no intersection of feature sets even if they have the same meaning. To obtain more reliable feature sets, we retrieve Web snippets by querying the phrase AND the *anchor* of the source phrase. The "anchored version" of Web snippets is retrieved in the following steps:

Step 2-1. Determine the anchor using Web snippets for the given source phrase. We regarded a noun which most frequently modifies the source phrase as its anchor. Examples of source phrases and their anchors are shown in (6).

Step 2-2. Retrieve Web snippets by querying the anchor for the source phrase AND each of source and target phrases, respectively.

Step 2-3. Extract features for HITS, BOW, MOD. Those sets are referred to as Anc.$*$, while the normal versions are referred to as Nor.$*$.

(6) a. "*emi:o:ukaberu*" $\cdots$ "*manmen*"
(be smiling $\cdots$ from ear to ear)
b. "*doriburu:de:kake:agaru*" $\cdots$ "*saido*"
(overlap by dribbling $\cdots$ side)
c. "*yoi:sutaato:o:kiru*" $\cdots$ "*saisaki*"
(make a good start $\cdots$ good sign)

3.3 Computing paraphrasability

Paraphrasability is finally computed by two conventional distributional similarity measures. The first is the measure proposed in (Lin and Pantel, 2001):

$$Par_{Lin}(s \Rightarrow t) = \frac{\sum_{f \in F_s \cap F_t} (w(s,f) + w(t,f))}{\sum_{f \in F_s} w(s,f) + \sum_{f \in F_t} w(t,f)},$$

where F_s and F_t denote feature sets for s and t, respectively. $w(x,f)$ stands for the weight (frequency in our experiment) of f in F_x.

While Par_{Lin} is symmetric, it has been argued that it is important to determine the direction of paraphrase. As an asymmetric measure, we examine α-skew divergence defined by the following equation (Lee, 1999):

$$d_{skew}(t,s) \quad = \quad D\left(P_s \| \alpha P_t + (1-\alpha) P_s\right),$$

where P_x denotes a probability distribution estimated[6] from a feature set F_x. How well P_t approximates P_s is calculated based on the KL divergence, D. The parameter α is set to 0.99, following tradition, because the optimization of α is difficult. To take consistent measurements, we define the paraphrasability score Par_{skew} as follows:

$$Par_{skew}(s \Rightarrow t) \quad = \quad \exp\left(-d_{skew}(t,s)\right).$$

[6]We estimate them simply using maximum likelihood estimation, i.e., $P_x(f) = w(x,f) / \sum_{f' \in F_x} w(x,f')$.

Table 1: # of sampled source phrases and automatically generated syntactic variants.

Phrase type	# of tokens	# of types	th	types	Cov.(%)	Output		Ave.
$N:C:V$	20,200,041	4,323,756	1,000	1,014	10.7	1,536	(489)	3.1
$N_1:N_2:C:V$	3,796,351	2,013,682	107	1,005	6.3	88,040	(966)	91.1
$N:C:V_1:V_2$	325,964	213,923	15	1,022	12.9	75,344	(982)	76.7
$N:C:Adv:V$	1,209,265	923,475	21	1,097	3.9	8,281	(523)	15.7
$Adj:N:C:V$	378,617	233,952	20	1,049	14.1	128	(50)	2.6
$N:C:Adj$	788,038	203,845	86	1,003	31.4	3,212	(992)	3.2
Total	26,698,276	7,912,633		6,190		176,541	(4,002)	44.1

Table 2: # of syntactic variants whose paraphrasability scores are computed.
Nor.HITS $\supset$ *Nor.BOW.*$*$ $\supset$ *Nor.MOD.*$*$. *Anc.HITS* $\supset$ *Anc.BOW.*$*$ $\supset$ *Anc.MOD.*$*$.
Nor.HITS $\supset$ *Anc.HITS*. *Nor.BOW.*$*$ $\supset$ *Anc.BOW.*$*$. *Nor.MOD.*$*$ $\supset$ *Anc.MOD.*$*$. X denotes the set of syntactic variants whose scores are computed based on X.

	Nor.HITS			Nor.BOW.$*$			Nor.MOD.$*$			Anc.HITS			Anc.BOW.$*$			Anc.MOD.$*$			Mainichi		
Phrase type	Output		Ave.	Output		Ave.	Output		Ave.	Output		Ave.	Output		Ave.	Output		Ave.	Output		Ave.
$N:C:V$	1,405	(489)	2.9	1,402	(488)	2.9	1,396	(488)	2.9	1,368	(488)	2.8	1,366	(487)	2.8	1,360	(487)	2.8	1,103	(457)	2.4
$N_1:N_2:C:V$	9,544	(964)	9.9	9,249	(922)	10.0	8,652	(921)	9.4	7,437	(897)	8.3	7,424	(894)	8.3	6,795	(891)	7.6	3,041	(948)	3.2
$N:C:V_1:V_2$	3,769	(876)	4.3	3,406	(774)	4.4	3,109	(762)	4.1	2,517	(697)	3.6	2,497	(690)	3.6	2,258	(679)	3.3	1,156	(548)	2.1
$N:C:Adv:V$	690	(359)	1.9	506	(247)	2.0	475	(233)	2.0	342	(174)	2.0	339	(173)	2.0	322	(168)	1.9	215	(167)	1.3
$Adj:N:C:V$	45	(20)	2.3	45	(20)	2.3	42	(17)	2.5	41	(18)	2.3	41	(18)	2.3	39	(16)	2.4	14	(7)	2.0
$N:C:Adj$	1,459	(885)	1.6	1,459	(885)	1.6	1,399	(864)	1.6	1,235	(809)	1.5	1,235	(809)	1.5	1,161	(779)	1.5	559	(459)	1.2
Total	16,912	(3,593)	4.7	16,067	(3,336)	4.8	15,073	(3,285)	4.6	12,940	(3,083)	4.2	12,902	(3,071)	4.2	11,935	(3,020)	4.0	6,088	(2,586)	2.4

Now Par_x falls within $[0, 1]$, and a larger Par_x indicates a more paraphrasable pair of phrases.

4 Experimental setting

We conduct empirical experiments to evaluate the proposed methods. Settings are described below.

4.1 Test collection

First, source phrases were sampled from a 15 years of newspaper articles (Mainichi 1991-2005, approximately 1.5GB). Referring to the dependency structure given by CaboCha, we extracted most frequent 1,000+ phrases for each of 6 phrase types. These phrases were then fed to a system proposed in (Fujita et al., 2007) to generate syntactic variants. The numbers of the source phrases and their syntactic variants are summarized in Table 1, where the numbers in the parentheses indicate that of source phrases paraphrased. At least one candidate was generated for 4,002 (64.7%) phrases. Although the system generates numerous syntactic variants from a given phrase, most of them are erroneous. For example, among 159 syntactic variants that are automatically generated for the phrase "*songai:baishou:o:motomeru*" (demand compensation for damages), only 8 phrases are grammatical, and only 5 out of 8 are correct paraphrases.

Paraphrasability of each pair of source phrase and candidate is then computed by the methods proposed in Section 3. Table 2 summarizes the numbers of pairs whose features can be extracted from the Web snippets. While more than 90% of candidates were discarded due to 'No hits' in the Web, at least one candidate survived for 3,020 (48.8%) phrases. Mainichi is a baseline which counts HITS in the corpus used for sampling source phrases.

4.2 Samples for evaluation

We sampled three sets of pairs for evaluation, where Mainichi, $*$.HITS, $*$.BOW, $*$.MOD, the harmonic mean of the scores derived from $*$.BOW and $*$.MOD (referred to as $*$.HAR), and two distributional similarity measures for $*$.BOW, $*$.MOD, and $*$.HAR, in total 15 models, are compared.

Ev.Gen: This investigates how well a correct candidate is ranked first among candidates for a given phrase using the top-ranked pairs for randomly sampled 200 source phrases for each of 15 models.

Ev.Rec: This assesses how well a method gives higher scores to correct candidates using the 200-best pairs for each of 15 models.

Ev.Ling: This compares paraphrasability of each phrase type using the 20-best pairs for each of 6 phrase type and 14 Web-based models.

4.3 Criteria of paraphrasability

To assess by human the paraphrasability discussed in Section 3, we designed the following four questions based on (Szpektor et al., 2007):

Q_{sc}: Is s a correct phrase in Japanese?

Q_{tc}: Is t a correct phrase in Japanese?

Q_{s2t}: Does t hold if s holds and can t substituted for s in some context?

Q_{t2s}: Does s hold if t holds and can s substituted for t in some context?

5 Experimental results

5.1 Agreement of human judge

Two human assessors separately judged all of the 1,152 syntactic variant pairs (for 962 source phrases) within the union of the three sample sets. They agreed on all four questions for 795 (68.4%) pairs. For the 963 (83.6%) pairs that passed Q_{sc} and Q_{tc} in both two judges, we obtained reasonable agreement ratios 86.9% and 85.0% and substantial Kappa values 0.697 and 0.655 for assessing Q_{s2t} and Q_{t2s}.

5.2 Ev.Gen

Table 3 shows the results for Ev.Gen, where the *strict precision* is calculated based on the number of two positive judges for Q_{s2t}, while the *lenient precision* is for at least one positive judge for the same question. $*$.MOD and $*$.HAR outperformed the other models, although there was no statistically significant difference[7]. Significant differences between Mainichi and the other models in lenient precisions indicate that the Web enables us to compute paraphrasability more accurately than a limited size of corpus.

From a closer look at the distributions of paraphrasability scores of $*$.BOW and $*$.MOD shown in Table 4, we find that if a top-ranked candidate for a given phrase is assigned enough high score, it is very likely to be correct. The scores of Anc.$*$ are distributed in a wider range than those of Nor.$*$, preserving precision. This allows us to easily skim the most reliable portion by setting a threshold.

5.3 Ev.Rec

The results for Ev.Rec, as summarized in Table 5, show the significant differences of performances between Mainichi or $*$.HITS and the other models. The results of $*$.HITS supported the importance of comparing features of phrases. On the other hand, $*$.BOW performed as well as $*$.MOD and $*$.HAR. This sounds nice because BOW features can be extracted extremely quickly and accurately.

Unfortunately, Anc.$*$ led only a small impact on strict precisions. We speculate that the selection of the anchor is inadequate. Another possible interpretation is that source phrases are rarely ambiguous, because they contain at least two content words. In paraphrase generation, capturing the correct boundary of phrases is rather vital, because the source phrase is usually assumed to be grammatical. Q_{sc} for 55 syntactic variants (for 44 source phrases) were actually judged incorrect.

The lenient precisions, which were reaching a ceiling, implied the limitation of the proposed methods. Most common errors among the proposed methods were generated by a transformation pattern $N_1 : N_2 : C : V \Rightarrow N_2 : C : V$. Typically, dropping a nominal element N_1 of the given nominal compound $N_1 : N_2$ generalizes the meaning that the compound conveys, and thus results correct paraphrases. However, it caused errors in some cases; for example, since N_1 was the semantic head in (7), dropping it was incorrect.

(7) s. "*shukketsu:taryou:de:shibou-suru*" (die due to heavy blood loss)
t.*"*taryou:de:shibou-suru*" (die due to plenty)

[7] $p < 0.05$ in 2-sample test for equality of proportions.

Table 3: Precision for 200 candidates (Ev.Gen).

	Strict		Lenient	
Model	Nor.$*$	Anc.$*$	Nor.$*$	Anc.$*$
Mainichi	77 (39%)	- -	101 (51%)	- -
HITS	84 (42%)	83 (42%)	120 (60%)	119 (60%)
BOW.Lin	82 (41%)	85 (43%)	123 (62%)	124 (62%)
BOW.skew	86 (43%)	87 (44%)	125 (63%)	124 (62%)
MOD.Lin	91 (46%)	91 (46%)	130 (65%)	131 (66%)
MOD.skew	92 (46%)	90 (45%)	132 (66%)	130 (65%)
HAR.Lin	90 (45%)	90 (45%)	129 (65%)	130 (65%)
HAR.skew	93 (47%)	90 (45%)	134 (67%)	131 (66%)

Table 4: Distribution of paraphrasability scores and lenient precision (Ev.Gen).

	Nor.BOW		Anc.BOW	
$Par(s \Rightarrow t)$	Lin	skew	Lin	skew
0.9-1.0	11/ 12 (92%)	0/ 0 -	17/ 18 (94%)	2/ 2 (100%)
0.8-1.0	45/ 49 (92%)	1/ 1 (100%)	45/ 50 (90%)	6/ 6 (100%)
0.7-1.0	72/ 88 (82%)	7/ 7 (100%)	73/ 92 (79%)	10/ 11 (91%)
0.6-1.0	94/127 (74%)	11/ 11 (100%)	83/113 (74%)	12/ 13 (92%)
0.5-1.0	102/145 (70%)	13/ 13 (100%)	96/128 (75%)	14/ 15 (93%)
0.4-1.0	107/158 (68%)	13/ 14 (93%)	103/145 (71%)	21/ 22 (96%)
0.3-1.0	113/173 (65%)	25/ 26 (96%)	114/166 (69%)	31/ 32 (97%)
0.2-1.0	119/184 (65%)	40/ 41 (98%)	121/186 (65%)	49/ 50 (98%)
0.1-1.0	123/198 (62%)	74/ 86 (86%)	124/200 (62%)	82/ 99 (83%)
0.0-1.0	123/200 (62%)	125/200 (63%)	124/200 (62%)	124/200 (62%)
Variance	0.052	0.031	0.061	0.044
	Nor.MOD		Anc.MOD	
$Par(s \Rightarrow t)$	Lin	skew	Lin	skew
0.9-1.0	2/ 2 (100%)	0/ 0 -	7/ 7 (100%)	1/ 1 (100%)
0.8-1.0	10/ 10 (100%)	0/ 0 -	12/ 13 (92%)	2/ 2 (100%)
0.7-1.0	13/ 14 (93%)	0/ 0 -	17/ 18 (94%)	6/ 6 (100%)
0.6-1.0	20/ 21 (95%)	1/ 1 (100%)	27/ 28 (96%)	9/ 9 (100%)
0.5-1.0	31/ 32 (97%)	6/ 6 (100%)	36/ 37 (97%)	10/ 10 (100%)
0.4-1.0	42/ 44 (96%)	11/ 11 (100%)	51/ 53 (96%)	12/ 12 (100%)
0.3-1.0	61/ 68 (90%)	12/ 12 (100%)	61/ 68 (90%)	13/ 14 (93%)
0.2-1.0	81/ 92 (88%)	13/ 13 (100%)	82/ 94 (87%)	18/ 19 (95%)
0.1-1.0	105/133 (79%)	17/ 18 (94%)	104/126 (83%)	24/ 25 (96%)
0.0-1.0	130/200 (65%)	132/200 (66%)	131/200 (66%)	130/200 (65%)
Variance	0.057	0.014	0.072	0.030

Table 5: Precision for 200 candidates (Ev.Rec).

	Strict		Lenient	
Model	Nor.∗	Anc.∗	Nor.∗	Anc.∗
Mainichi	78 (39%)	- -	111 (56%)	- -
HITS	71 (36%)	93 (47%)	113 (57%)	128 (64%)
BOW.Lin	159 (80%)	162 (81%)	193 (97%)	191 (96%)
BOW.skew	154 (77%)	158 (79%)	192 (96%)	191 (96%)
MOD.Lin	158 (79%)	164 (82%)	192 (96%)	193 (97%)
MOD.skew	156 (78%)	161 (81%)	191 (96%)	191 (96%)
HAR.Lin	157 (79%)	164 (82%)	192 (96%)	194 (97%)
HAR.skew	155 (78%)	160 (80%)	191 (96%)	191 (96%)

Table 6: Precision for each phrase type (Ev.Ling).

Phrase type	Strict	Lenient
$N:C:V$	52/ 98 (53%)	69/ 98 (70%)
$N_1:N_2:C:V$	51/ 72 (71%)	64/ 72 (89%)
$N:C:V_1:V_2$	42/ 86 (49%)	60/ 86 (70%)
$N:C:Adv:V$	33/ 61 (54%)	44/ 61 (72%)
$Adj:N:C:V$	0/ 25 (0%)	4/ 25 (16%)
$N:C:Adj$	18/ 73 (25%)	38/ 73 (52%)
Total	196/415 (47%)	279/415 (67%)

Table 7: # of features.

	Nor.BOW	Nor.MOD	Anc.BOW	Anc.MOD
# of features (type)	73,848	471,720	72,109	409,379
average features (type)	1,322	211	1,277	202
average features (token)	4,883	391	4,728	383

5.4 Ev.Ling

Finally the results for Ev.Ling is shown in Table 6. Paraphrasability of syntactic variants for phrases containing an adjective was poorly computed. The primal source of errors for $Adj:N:C:V$ type phrases was the subtle change of nuance by switching syntactic heads as illustrated in (8), where underlines indicate heads.

(8) s. "*yoi:shigoto:o:suru*" (do a good job)
$t_1 \not\simeq$"*yoku:shigoto-suru*" (work hard)
$t_2 \not\simeq$"*shigoto:o:yoku:suru*" (improve the work)

Most errors in paraphrasing $N:C:Adj$ type phrases, on the other hand, were caused due to the difference of aspectual property and agentivity between adjectives and verbs. For example, ($9s$) can describe not only things those qualities have been improved as inferred by ($9t$), but also those originally having a high quality. Q_{s2t} for (9) was thus judged incorrect.

(9) s. "*shitsu:ga:takai*" (having high quality)
$t \not\simeq$"*shitsu:ga:takamaru*" (quality rises)

Precisions of syntactic variants for the other types of phrases were higher, but they tended to include trivial paraphrases such as shown in (10) and (11). Yet, collecting paraphrase instances statically will contribute to paraphrase recognition tasks.

(10) s. "*shounin:o:eru*" (clear)
t. "*shounin-sa-re-ru*" (be approved)

(11) s. "*eiga:o:mi:owaru*" (finish seeing the movie)
t. "*eiga:ga:owaru*" (the movie ends)

6 Discussion

As described in the previous sections, our quite naive methods have shown fairly good performances in this first trial. This section describes some remaining issues to be discussed further.

The aim of this study is to create a thesaurus of phrases to recognize and generate phrases that are semantically equivalent and syntactically substitutable, following the spirit described in (Fujita et al., 2007). Through the comparisons of Nor.∗ and Anc.∗, we have shown a little evidence that the ambiguity of phrases was not problematic at least for handling syntactic variants, arguing the necessity of detecting the appropriate phrase boundaries.

To overcome the data sparseness problem, Web snippets are harnessed. Features extracted from the snippets outperformed newspaper corpus; however, the small numbers of features for phrases shown in Table 7 and the lack of sophisticated weighting function suggest that the problem might persist. To examine the proposed features and measures further, we plan to use TSUBAKI[8], an indexed Web corpus developed for NLP research, because it allow us to obtain snippets as much as it archives.

The use of larger number of snippets increases the computation time for assessing paraphrasability. For reducing it as well as gaining a higher coverage, the enhancement of the paraphrase generation system is necessary. A look at the syntactic variants automatically generated by a system, which we proposed, showed that the system could generate syntactic variants for only a half portion of the input, producing many erroneous ones (Section 4.1). To prune a multitude of incorrect candidates, statistical language models such as proposed in (Habash, 2004) will be incorporated. In parallel, we plan to develop a paraphrase generation system which lets us to quit from the labor of maintaining patterns such as shown in (4). We think a more unrestricted generation algorithm will gain a higher coverage, preserving the meaning as far as handling syntactic variants of predicate phrases.

[8] http://tsubaki.ixnlp.nii.ac.jp/se/index.cgi

7 Conclusion

In this paper, we proposed a method of assessing paraphrasability between automatically generated syntactic variants of predicate phrases. Web snippets were utilized to overcome the data sparseness problem, and the conventional distributional similarity measures were employed to quantify the similarity of feature sets for the given pair of phrases. Empirical experiments revealed that features extracted from the Web snippets contribute to the task, showing promising results, while no significant difference was observed between two measures.

In future, we plan to address several issues such as those described in Section 6. Particularly, at present, the coverage and portability are of our interests.

Acknowledgments

We are deeply grateful to all anonymous reviewers for their valuable comments. This work was supported in part by MEXT Grant-in-Aid for Young Scientists (B) 18700143, and for Scientific Research (A) 16200009, Japan.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 597–604.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 50–57.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL*), pages 16–23.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics* (*COLING*), pages 350–356.

Mark Dras. 1999. *Tree adjoining grammar and the reluctant paraphrasing of text*. Ph.D. thesis, Division of Information and Communication Science, Macquarie University.

Atsushi Fujita, Shuhei Kato, Naoki Kato, and Satoshi Sato. 2007. A compositional approach toward dynamic phrasal thesaurus. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing* (*WTEP*), pages 151–158.

Maayan Geffet and Ido Dagan. 2004. Feature vector quality and distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics* (*COLING*), pages 247–253.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 107–116.

Nizar Habash. 2004. The use of a structural N-gram language model in generation-heavy hybrid machine translation. In *Proceedings of the 3rd International Natural Language Generation Conference* (*INLG*), pages 61–69.

Zellig Harris. 1957. Co-occurrence and transformation in linguistic structure. *Language*, 33(3):283–340.

Zellig Harris. 1968. *Mathematical structures of language*. John Wiley & Sons.

Christian Jacquemin. 1999. Syntagmatic and paradigmatic representations of term variation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 341–348.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 25–32.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Igor Mel'čuk and Alain Polguère. 1987. A formal lexicon in meaning-text theory (or how to do lexica with words). *Computational Linguistics*, 13(3-4):261–275.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL*), pages 102–109.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning inferential selectional preferences. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics* (*NAACL-HLT*), pages 564–571.

Satoshi Sekine. 2005. Automatic paraphrase discovery based on context and keywords between NE pairs. In *Proceedings of the 3rd International Workshop on Paraphrasing* (*IWP*), pages 80–87.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling Web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pages 41–48.

Idan Szpektor, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 456–463.

Kentaro Torisawa. 2006. Acquiring inference rules with temporal constraints by using Japanese coordinated sentences and noun-verb co-occurrences. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL*), pages 57–64.

Julie Weeds, David Weir, and Bill Keller. 2005. The distributional similarity of sub-parses. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 7–12.

Hua Wu and Ming Zhou. 2003. Synonymous collocation extraction using translation information. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (*ACL*), pages 120–127.

Augmenting Wikipedia with Named Entity Tags

Wisam Dakka
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
wisam@cs.columbia.edu

Silviu Cucerzan
Microsoft Research
1 Microsoft Way
Redmond, WA 98052
silviu@microsoft.com

Abstract

Wikipedia is the largest organized knowledge repository on the Web, increasingly employed by natural language processing and search tools. In this paper, we investigate the task of labeling Wikipedia pages with standard named entity tags, which can be used further by a range of information extraction and language processing tools. To train the classifiers, we manually annotated a small set of Wikipedia pages and then extrapolated the annotations using the Wikipedia category information to a much larger training set. We employed several distinct features for each page: bag-of-words, page structure, abstract, titles, and entity mentions. We report high accuracies for several of the classifiers built. As a result of this work, a Web service that classifies any Wikipedia page has been made available to the academic community.

1 Introduction

Wikipedia, one of the most frequently visited web sites nowadays, contains the largest amount of knowledge ever gathered in one place by volunteer contributors around the world (Poe, 2006). Each Wikipedia article contains information about one entity or concept, gathers information about entities of one particular type of entities (the so-called *list pages*), or provides information about homonyms (*disambiguation pages*). As of July 2007, Wikipedia contains close to two million articles in English. In addition to the English-language version, there are 200 versions in other languages. Wikipedia has about 5 million registered contributors, averaging more than 10 edits per contributor.

Natural language processing and search tools can greatly benefit from Wikipedia by using it as an authoritative source of common knowledge and by exploiting its interlinked structure and disambiguation pages, or by extracting concept co-occurrence information. This paper presents a successful study on enriching the Wikipedia data with named entity tags. Such tags could be employed by disambiguation systems such as Bunescu and Paşca (2006) and Cucerzan (2007), in mining relationships between named entities, or in extracting useful facet terms from news articles (e.g., Dakka and Ipeirotis, 2008).

In this work, we classify the Wikipedia pages into categories similar to those used in the CoNLL shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) and ACE (Doddington et al., 2004). To the best of our knowledge, this is the first attempt to perform such classification on the English language version of the collection. [1] Although the task settings are different, the results we obtained are comparable with those previously reported in document classification tasks.

We examined the Wikipedia pages to extract several feature groups for our classification task. We also observed that each entity/concept has at least two pseudo-independent views (page-based features and link-based features), which allow the use a co-training method to boost the performance of classifiers trained separately on each view.

The classifier that achieved the best accuracy on out test set was applied then to all Wikipedia pages and its classifications are provided to the academic community for use in future studies through a Web service.[2]

[1] Watanabe et al. (2007) have reported recently experiments on categorizing named entities in the Japanese version of Wikipedia using a graph-based approach.
[2] The Web service is available at *wikinet.stern.nyu.edu*.

2 Related Work

This study is related to the area of named entity recognition, which has supported extensive evaluations (CoNLL and ACE). Since the introduction of this task in MUC-6 (Grishman and Sundheim, 1996), numerous systems using various ways of exploiting entity-specific and local context features were proposed, from relatively simple character-based models such as Cucerzan and Yarowsky (2002) and Klein et al. (2003) to complex models making use of various lexical, syntactic, morphological, and orthographical information, such as Wacholder et al. (1997), Fleischman and Hovy (2002), and Florian et al. (2003). While the task we address is not the conventional named entity recognition but rather document classification, our classes are a derived from the labels traditionally employed in named entity recognition, following the CoNLL and ACE guidelines, as described in Section 3.

The areas of text categorization and document classification have also been extensively researched over time. These task have the goal of assigning to each document in a collection one or several labels from a given set, such as Newsgroups (Lang, 1995), Reuters (Reuters, 1997), Yahoo! (Mladenic, 1998), Open Directory Project (Chakrabarti et al., 2002), and Hoover's Online (Yang et al., 2002). Various supervised machine learning algorithms have been applied successfully to the document classification problem (e.g., Joachims, 1999; Quinlan, 1993; Cohen, 1995). Dumais et al. (1998) and Yang and Liu (1999) reported that support vector machines (SVM) and K-Nearest Neighbor performed the best in text categorization. We adopted SVM as our algorithm of choice because of these findings and also because SVMs have been shown robust to noise in the feature set in several studies. While Joachims (1998) and Rogati and Yang (2002) reported no improvement in SVM performance after applying a feature selection step, Gabrilovich and Markovitch (2004) showed that for collection with numerous redundant features, aggressive feature selection allowed SVMs to actually improve their performance. However, performing an extensive investigation of classification performance across various machine learning algorithms has been beyond the purpose of this work, in which we ran classification experiments using SVMs and compared them only with the results of similar systems employing Naïve Bayes.

In addition to the traditional bag-of-words, which has been extensively used for the document classification task (e.g. Sebastiani, 2002), we employed various other Wikipedia-specific feature sets. Some of these have been previously employed for various tasks by Gabrilovich and Markovitch, (2006); Overell and Ruger (2006), Cucerzan (2007), and Suchanek et al. (2007).

3 Classifying Wikipedia Pages

The Wikipedia pages that we analyzed in this study can be divided into three types:

Disambiguation Page (DIS): is a special kind of page that usually contains the word "disambiguation" in its title, and that contains several possible disambiguations of a term.

Common Page (COMM): refers to a common object rather than a named entity. Generally, if the name of an object or concept appears non-capitalized in text then it is very likely that the object or the concept is of common nature (heuristic previously employed by Bunescu and Paşca, 2006). For example, the Wikipedia page "Guitar" refers to a common object rather than a named entity.

Named Entity Page: refers to a specific object or set of objects in the world, which is/are commonly referred to using a certain proper noun phrase. For example, any particular person is a named entity, though the concept of "people" is not a named entity. Note that most names are ambiguous. "Apollo" can refer to more than 30 different entities of different types, for example, the Finnish rock band of the late 1960s/early 1970s , the Greek god of light, healing, and poetry, and the series of space missions run by NASA.

To classify the named entities in Wikipedia, we adopted a restricted version of the ACE guidelines (ACE), using four main entity classes (also similar to the classes employed in the CoNLL evaluations):

Animated Entities (PER): An animate entity can be either of type human or non-human. **Human** entities are either humans that are known to have lived (e.g., "Leonardo da Vinci", "Britney Spears", "Gotthard of Hildesheim", "Saint Godehard") or humanoid individuals in fictional works, such as books, movies, TV shows, and comics (e.g., "Harry Potter", "Batman", "Sonny"

the robot from the movie "I, Robot"). Fictional characters also include mythological figures and deities (e.g. "Zeus", "Apollo", "Jupiter"). The fictional nature of a character must be explicitly indicated. **Non-human** entities are any particular animal or alien that has lived or that is described in a fictional work and can be singled out using a name.

Organization Entities (ORG): An organization entity must have some formally established association. Typical examples are businesses (e.g., "Microsoft", "Ford"), governmental bodies (e.g., "United States Congress"), non-governmental organizations (e.g., "Republican Party", "American Bar Association"), science and health units (e.g., "Massachusetts General Hospital"), sports organizations and teams (e.g., "Angolan Football Federation", "San Francisco 49ers"), religious organizations (e.g., "Church of Christ"), and entertainment organizations, including formally organized music groups (e.g., "San Francisco Mime Troupe", the rock band "The Police"). Industrial sectors and industries (e.g., "Petroleum industry") are also treated as organization entities, as well as all media and publications.

Location Entities (LOC): These are physical locations (regions in space) defined by geographical, astronomical, or political criteria. They are of three types: **Geo-Political** entities are composite entities comprised of a physical location, a population, a government, and a nation (or province, state, county, city, etc.). A Wikipedia page that mentions all these components should be labeled as Geo-Political Entity (e.g., "Hawaii", "European Union", "Australia", and "Washington, D.C."). **Locations** are places defined on a geographical or astronomical basis and do not constitute a political entity. These include mountains, rivers, seas, islands, continents (e.g., "the Solar system", "Mars", "Hudson River", and "Mount Rainier"). **Facilities** are artifacts in the domain of architecture and civil engineering, such as buildings and other permanent man-made structures and real estate improvements: airports, highways, streets, etc.

Miscellaneous Entities (MISC): About 25% of the named entities in Wikipedia are not of the types listed above. By examining several hundred examples, we concluded that the majority of these named entities can be classified in one of the following classes: **Events** refer to historical events or actions with some certain duration, such as wars, sport events, and trials (e.g., "Gulf War", "2006 FIFA World Cup", "Olympic Games", "O.J. Simpson trial"). **Works of art** refer to named works that are imaginative in nature. Examples include books, movies, TV programs, etc. (e.g., the "Batman" movie, "The Tonight Show", the "Harry Potter" books). **Artifacts** refer to man-made objects or products that have a name and cannot generally be labeled as art. This includes mass-produced merchandise and lines of products (e.g. the camera "Canon PowerShot Pro1", the series "Canon PowerShot", the type of car "Ford Mustang", the software "Windows XP"). Finally **Processes** include all named physical and chemical processes (e.g., "Ettinghausen effect"). Abstract formulas or algorithms that have a name are also labeled as processes (e.g., "Naive Bayes classifier").

4 Features Used. Independent Views

When creating a Wikipedia page and introducing a new entity, contributors can refer to other related Wikipedia entities, which may or may not have corresponding Wikipedia pages. This way of generating content creates an internal web graph and, interesting, results in the presence of two different and pseudo-independent views for each entity. We can represent an entity using the content written on the entity page, or alternatively, using the context from a reference on the related page. For example, Figures 1 and 2 show the two independent views of the entity "Gwen Stefani".

1 such as 'Let Me Blow Ya Mind' by Eve and [[Gwen Stefani]] (whom he would produce
2 In the video "[[Cool (song)—Cool]]", [[Gwen Stefani]] is made-up as Monroe.
3 '[[South Side (song)—South Side]]' (featuring [[Gwen Stefani]]) #14 US
4 [[1969]] - [[Gwen Stefani]], American singer ([[No Doubt]])
5 [[Rosie Gaines]], [[Carmen Electra]], [[Gwen Stefani]], [[Chuck D]], [[Angie Stone]],
6 In late [[2004]], [[Gwen Stefani]] released a hit song called 'Rich Girl' which
7 [[Gwen Stefani]] - lead singer of the band [[No Doubt]], who is now a successful
8 [[Social Distortion]], and [[TSOL]]. [[Gwen Stefani]], lead vocalist of the [[alternative rock]]
9 main proponents (along with [[Gwen Stefani]] and [[Ashley Judd]]) in bringing back the
10 The [[United States—American]] singer [[Gwen Stefani]] references Harajuku in several

Figure 1. A partial list of contextual references taken from Wikipedia for the named entity "Gwen Stefani". (There are over 600 such references.)

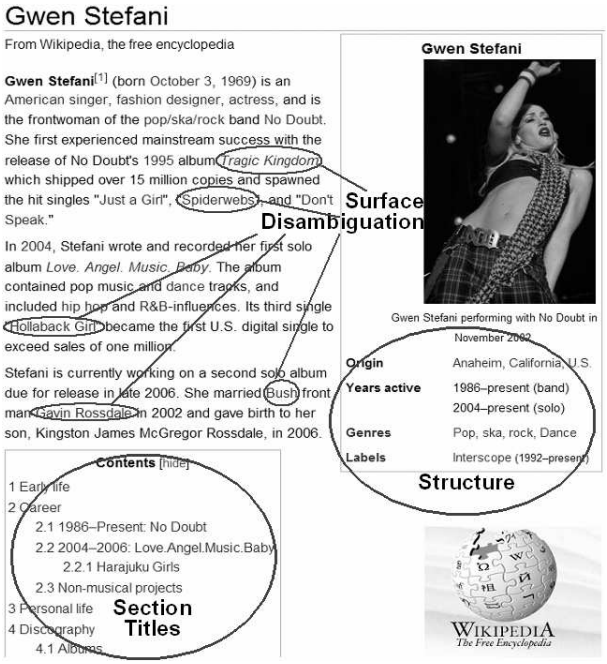


Figure 2. Wikipedia page for the named entity "Gwen Stefani". Other than the regular text, information such as surface and disambiguated entities, structure properties, and section titles can be easily extracted.

We utilize this important observation to extract our features based on these two independent views: page-based features and context features. We discuss these in greater detail next.

4.1 Page-Based Features

A typical Wikipedia page is usually written and edited by several contributors. Each page includes a rich set of information including the following elements: titles, section titles, paragraphs, multimedia objects, hyperlinks, structure data, surface entities and their disambiguations. Figure 2 shows some of these elements in the page dedicated to singer "Gwen Stefani". We use the Wikipedia page XML syntax to draw a set of different page-based feature vectors, including the following:

Bag of Words (BOW): This vector is the term frequency representation of the entire page.

Structured Data (STRUCT): Many Wikipedia pages contain useful data organized in tables and other structural representations. In Figure 2, we see that contributors have used a table representation to list different properties about Gwen Stefani. We extract for each page, using the Wikipedia syntax, the bag-of-words feature vector that corresponds to this structured data only.

<abstract>
Gwen Rene StefaniSome sources give Stefani's first name as Gwendolyn, but her first name is simply Gwen. Her listing on the California Birth Index from the Center for Health Statistics gives a birth name of Gwen Rene Stefani.
</abstract>

Figure 3. The abstract provided by Wikipedia for "Gwen Stefani". Note the concatenation of "Stefani" and "Some", which results in a new word, and is a relevant example of noise encountered in Wikipedia text.

First Paragraph (FPAR): We examined several hundred pages, and observed that a human could label most of the pages by reading only the first paragraph. Therefore, we built the feature vector that contains the bag-of-word representation of the page's first paragraph.

Abstract (ABS): For each page, Wikipedia provides a summary of several lines about the entity described on the page. We use this summary to draw another bag-of-word feature vector based on the provided abstracts only. For example, Figure 3 shows the abstract for the entity "Gwen Stefani".

Surface Forms and Disambiguations (SFD): Contributors use the Wikipedia syntax to link from one entity page to another. In the page of Figure 2, for example, we have references to several other Wikipedia entities, such as "hip hop", "R&B", and "Bush". Wikipedia page syntax lets us extract the disambiguated meaning of each of these references, which are "Hip hop music," "Rhythm and blues," and "Bush band", respectively. For each page, we extract all the surface forms used by contributors in text (such as "hip hop") and their disambiguated meanings (such as "Hip hop music"), and build feature vectors to represent them.

4.2 Context Features

Figure 1 shows some of the ways contributors to Wikipedia refer to the entity "Gwen Stefani". The Wikipedia version that we analyzed contains about 35 million references to entities in the collection. On average, each page has five references to other entities.

We decided to make use of the text surrounding these references to draw contextual features, which can capture both syntactic and semantic properties of the referenced entity. For each entity reference, we compute the feature vectors by using a text window of three words to the left and to the right of the reference.

BOW	1,821,966	ABS	372,909
SFD	847,857	BCON	35,178,120
STRUCT	159,645	FPAR	781,938

Table 1. Number of features in each group, as obtained by examining all the Wikipedia pages.

We derived a unigram context model and a bigram context model, following the findings of previous work that such models benefit from employing information about the position of words relative to the targeted term:

Unigram Context (UCON): The feature vector is constructed in a way that preserves the positional information of words in the context. Each feature f^t_i in the vector represents the total number of times a term *t* appears in position *i* around the entity.

Bigram Context (BCON): The bigram-based context model was built in a similar way to UCON, so that relative positional information is preserved.

5 Challenges

For our classification task, we faced several challenges. First, many Wikipedia entities have only a partial list of the feature groups discussed above. For example, contributors may refer to entities that do not exist in Wikipedia but might be added in the future. Also, not all the page-based features groups are available for every entity page. For instance, abstracts and structure features are only available for 68% and 79% of the pages, respectively. Second, we only had available several hundred labeled examples (as described in Section 6.1). Third, the feature space is very large compared to the typical text classification problem (see Table 1), and a substantial amount of noise plagues the data. A further investigation revealed that the difference in the dimensionality compared to text classification stems from the way Wikipedia pages are created: contributors make spelling errors, introduce new words, and frequently use slang, acronyms, and other languages than English.

We utilize all the features groups described in Section 4 and various combinations of them. This provides us with greater flexibility to use classifiers trained on different feature groups when Wikipedia entities miss certain types of features.

In addition, we try to take advantage of the independent views of each entity by employing a co-training procedure (Blum and Mitchell, 1998; Nigam and Ghani, 2000). In previous work, this has been shown to boost the performance of the weak classifiers on certain feature groups. For example, it is interesting to determine whether we can use the STRUCT view of a Wikipedia pages to boost the performance of the classifiers based on context. Alternatively, we can employ co-training on the STRUCT and SFD features, hypothesized as two independent views of the data.

6 Experiments and Findings

6.1 Training Data

We experimented with two data sets: **Human Judged Data** (HJD): This set was obtained in an annotation effort that followed the guidelines presented in Section 3. Due to the cost of the labeling procedure, this set was limited to a small random set of 800 Wikipedia pages. **Human Judged Data Extended** (HJDE): The initial classification results obtained using a small subset of HJD hinted to the need for more training data. Therefore, we devised a procedure that takes advantage of the fact that Wikipedia contributors have assigned many of the pages to one or more lists. For example, the page "List of novelists" contains a reference to "Orhan Pamuk", which is part of the HJD and is labeled as PER. Our extension procedure first uses the pages in the training set from HJD to extract the lists in Wikipedia that contain references to them and then projects the entity labels of the seeds to all elements in the lists. Unfortunately, not all the Wikipedia lists contain only references named entities of the same category. Furthermore, some lists are hierarchical and include sub-lists of different classes. To overcome these issues, we examined only leaf lists and manually filtered all the lists that by definition could have pages of different categories. Finally, we filtered out all list pages that contain entities in two or more entity classes (as described in Section 3).

Our partially manual extension procedure is as follows: 1) Pick a random sample of 400 entities from HJD along with their human judged labels; 2) Extract all the lists that contain any entity from this labeled sample; 3) Filter out the lists that contain entities from different entity classes (PER, ORG, LOC, MISC, and COM); 4) propagate the entity labels of the known entities in the lists to the other referenced entities; 5) Choose a random sample from all labeled pages with respect to the entity class distribution observed in HJD**.**

PER	MISC	ORG	LOC	COMM
41%	25.1%	11.2%	11.7%	11%

Table 2. The distribution of labels in the HJDE data set.

Our extension procedure resulted initially in 770 lists, which were then reduced to 501. In step (5), we chose a maximal random sample from all labeled pages in HJDE so that it matched the entity class distribution in the original HJD training set (shown in Table 2).

6.2 Classification

From the numerous machine learning algorithms available for our classification task (e.g., Joachims, 1999; Quinlan, 1993; Cohen, 1995), we chose to the SVMs (Vapnik, 1995), and the Naïve Bayes (John and Langley, 1995) algorithms because both can output probability estimates for their predictions, which are necessary for the co-training procedure. We use an implementation of SVM (Platt, 1999) with linear kernels and the Naïve Bayes implementation from the machine learning toolkit Weka3. Our implementation of co-training followed that of Nigam and Ghani (2000).

Using the HJDE data, we experimented with learning a classifier for each feature group discussed in Section 4. We report the results for two classification tasks: binary classification to identify all the Wikipedia pages of type PER, and 5-fold classification (PER, COM, ORG, LOC, and MISC).

To reduce the feature space, we built a term frequency dictionary taken from one year's worth of news data and restrict our feature space to contain only terms with frequency values higher than 10.

6.3 Results on Bag-of-words

This feature group is of particular interest, since it has been widely used for document classification and also, because every Wikipedia page has a BOW representation. We experimented with the two classification tasks for this feature group. For the binary classification task, both SVM and Naïve Bayes performed remarkably well, obtaining accuracies of 0.962 and 0.914, respectively. Table 3 shows detailed performance numbers for SVM and Naïve Bayes for the multi-class task. Unlike in the binary case, Naïve Bayes falls short of achieving results similar to those from SVM, which obtains an average F-measure of 0.928 and an average precision of 0.931.

	Precision		Recall		F-measure	
	SVM	NB	SVM	NB	SVM	NB
PER	0.944	0.918	0.959	0.771	0.951	0.838
MISC	0.927	0.824	0.920	0.687	0.924	0.750
ORG	0.940	0.709	0.928	0.701	0.934	0.705
LOC	0.958	0.459	0.949	0.863	0.954	0.599
COMM	0.887	0.680	0.869	0.714	0.878	0.697

Table 3. Precision, recall, and F1 measure for the multi-class classification task. Results are obtained using SVM and Naïve Bayes after a stratified cross-validation using HJDE data set and the bag-of-words features.

SFD	83.14%	ABS	68.96%
STRUCT	79.55%	BCON	83.57%

Table 4. Percentage of available examples HJDE for each feature group.

	Precision		Recall		F-measure	
	SVM	NB	SVM	NB	SVM	NB
BOW	**0.901**	0.858	0.894	0.880	**0.897**	0.869
SFD	0.851	0.775	0.830	0.882	0.840	0.825
STRUCT	**0.888**	0.840	0.875	0.856	**0.881**	0.848
FPAR	0.867	0.872	0.854	0.896	0.860	**0.884**
ABS	0.861	0.833	0.852	0.885	0.857	0.858
BCON	0.311	0.245	0.291	0.334	0.300	0.283

Table 5. Average precision, recall, and F1 measure values for the multi-class task. Results are obtained using SVM and Naïve Bayes across the different feature groups on the test set of HJDE.

6.4 Results on Other Feature Groups

We present now the results obtained using other groups of features. We omit the results on UCON due to their similarity with BCON. Recall that these features may not be present in all Wikipedia pages. Table 4 shows the availability of these features in the HJDE set. The lack of one feature group has a negative impact on the results of the corresponding classifier, as shown in Table 5. Noticeably, the results of the STRUCT features are very encouraging and confirm our hypothesis that such features are distinctive in identifying the type of the page. While results using STRUCT and FPAR are high, they are lower than the results obtained on BOW. In general, using SVM with BOW performed better than any other feature set, averaging 0.897 F-measure on test set. This could be because when using BOW, we have a larger training set than any other feature group. SVM with STRUCT and Naïve Bayes with FPAR performed

second and third best, with average F1 measure values of 0.881 and 0.860, respectively. The results also show that it is difficult to learn if a page is COMM in all learning combination. This could be related to the membership complexity of that class. Finally, the results on the bigram contextual features, namely BCON, for both SVM and Naïve Bayes are not encouraging and surprisingly low.

6.5 Results for Co-training

Motivated by the fact that some feature groups can be seen as independent views of the data, we used a co-training procedure to boost the classification accuracy. One combination of views that we examined is BCON with BOW, hoping to boost the classification performance of the bigram context features, as this classifier could be used for entities in any new text, not only for Wikipedia pages . Unfortunately, the results were not encouraging in either of the cases (SVM and Naïve Bayes) and for none of the other feature groups used instead of BOW. This indicates that the context features extracted have limited power and that further investigation of extracting relevant context features from Wikipedia is necessary.

7 Conclusions and Future Work

In this paper, we presented a study on the classification of Wikipedia pages with named entity labels. We explored several alternatives for extracting useful page-based and context-based features such as the traditional bag-of-words, page structure, hyperlink text, abstracts, section titles, and *n*-gram contextual features. While the classification with page features resulted in high classification accuracy, context-based and structural features did not work similarly well, either alone or in a co-training setup. This motivates future work to extract better such features. We plan to examine employing more sophisticated ways both for extracting contextual features and for using the implicit Wikipedia graph structure in a co-training setup.

Recently, the Wikipedia foundation has been taken steps toward enforcing a more systematic way to add useful structured data on each page by suggesting templates to use when a new page gets added to the collection. This suggests that in a not-so-distant future, we may be able to utilize the structured data features as attribute-value pairs rather than as bags of words, which is prone to losing valuable semantic information.

Finally, we have applied our classifier to all Wikipedia pages to determine their labels and made these data available in the form of a Web service, which can positively contribute to future studies that employ the Wikipedia collection.

References

ACE Project. At *http://www.nist.gov/speech/history/index.htm*

Reuters-1997. 1997. Reuters-21578 text categorization test collection. At *http://www.daviddlewis.com/resources/testcollections/reuters21578*

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT'98*, pages 92–100.

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998. NYU: Description of the MENE named entity system as used in MUC. In *Proceedings of MUC-7*.

R. Bunescu and M. Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL-2006*, pages 9–16.

S. Chakrabarti, M.M. Joshi, K. Punera, and D.M. Pennock. 2002. The structure of broad topics on the web. In: *Proceedings of WWW '02*, pages 251–262.

W.W. Cohen. 1995. Fast effective rule induction. In *Proceedings of ICML'95*.

S. Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL 2007*, pages 708–716.

S. Cucerzan and D. Yarowsky. 2002. Language Independent NER using a Unified Model of Internal and Contextual Evidence, in *Proceedings of CoNLL 2002*, pages 171–174.

W. Dakka and P. G. Ipeirotis. 2008. Automatic Extraction of Useful Facet Terms from Text Documents. In *Proceedings of ICDE 2008 (to appear).*

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. ACE program – task definitions and performance measures. In *Proceedings of LREC*, pages 837–840.

S. Dumais, J. Platt, D. Heckerman, and M. Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of CIKM '98*, pages 148–155.

M. Fleischman and E. Hovy. 2002. Fine Grained Classification of Named Entities. In *Proceedings of COLING'02*, pages 267–273.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, Named Entity Recognition through Classifier Combination, in *Proceedings of CoNLL 2003*, pages 168–171.

E. Gabrilovich and S. Markovitch. 2004. Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with c4.5. In *Proceedings of ICML '04*, page 41.

E. Gabrilovich and S. Markovitch. 2006. Overcoming-the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of AAI 2006*.

R. Grishman and B. Sundheim. 1996. Message Understanding Conference - 6: A brief history. In *Proceedings of COLING*, 466-471.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML '98*, pages 137–142.

T. Joachims. 1999. Making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning*, pages 169–184.

G.H. John and P. Langley. 1995. Estimating continuous distributions in Bayesian classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345.

D. Klein, J. Smarr, H. Nguyen, and C. D. Manning. 2003. Named Entity Recognition with Character-Level Models, in *Proceedings of CoNLL 2003*.

K. Lang. 1995. NewsWeeder: Learning to filter netnews. In Proceedings of ICML'95, pages 331–339.

D. Mladenic. 1998. Feature subset selection in text learning. In *Proceedings of ECML '98*, pages 95–100.

K. Nigam and R. Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *Proceedings of CIKM'00*, pages 86–93.

S.E. Overell and S. Ruger. 2006. Identifying and grounding descriptions of places. In *Workshop on Geographic Information Retrieval, SIGIR 2006*.

J.C. Platt. 1999. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*, pages 185–208.

M. Poe. 2006. The hive: Can thousands of wikipedians be wrong? How an attempt to build an online encyclopedia touched off history's biggest experiment in collaborative knowledge. The Atlantic Monthly, September 2006.

J.R. Quinlan. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

M. Rogati and Y. Yang. 2002. High-performing feature selection for text classification. In *Proceedings of CIKM '02*, pages 659–661.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing. Surveys*, 34(1):1–47.

F.M. Suchanek, G. Kasneci, and G. Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of WWW 2007*.

E.F. Tjong Kim Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

E. F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158.

V.N. Vapnik. 1995. The nature of statistical learning theory. Springer-Verlag New York, Inc.

N. Wacholder., Y. Ravin, and M. Choi. 1997. Disambiguation of proper names in text. In *Proceedings of ANLP'97*, pages 202-208.

Y. Watanabe, M. Asahara, and Y. Matsumoto. 2007. A Graph-based Approach to Named Entity Categorization in Wikipedia using Conditional Random Fields. In *Proc. of EMNLP-CoNLL 2007*, pages 649-657.

Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *Proceedings of SIGIR '99*, pages 42–49.

Y. Yang, S. Slattery, and R. Ghani. 2002. A study of approaches to hypertext categorization. *Journal of. Intelligent. Information. Systems*., 18(2-3):219–241.

Context Feature Selection for Distributional Similarity

Masato Hagiwara, Yasuhiro Ogawa, and Katsuhiko Toyama
Graduate School of Information Science,
Nagoya University
Furo-cho, Chikusa-ku, Nagoya, JAPAN 464-8603
{hagiwara, yasuhiro, toyama}@kl.i.is.nagoya-u.ac.jp

Abstract

Distributional similarity is a widely used concept to capture the semantic relatedness of words in various NLP tasks. However, accurate similarity calculation requires a large number of contexts, which leads to impractically high computational complexity. To alleviate the problem, we have investigated the effectiveness of automatic context selection by applying feature selection methods explored mainly for text categorization. Our experiments on synonym acquisition have shown that while keeping or sometimes increasing the performance, we can drastically reduce the unique contexts up to 10% of the original size. We have also extended the measures so that they cover context categories. The result shows a considerable correlation between the measures and the performance, enabling the automatic selection of effective context categories for distributional similarity.

1 Introduction

Semantic similarity of words is one of the most important lexical knowledge for NLP tasks including word sense disambiguation and synonym acquisition. To measure the semantic relatedness of words, a concept called *distributional similarity* has been widely used. Distributional similarity represents the relatedness of two words by the commonality of contexts the words share, based on the *distributional hypothesis* (Harris, 1985), which states that semantically similar words share similar contexts.

A wide range of contextual information, such as surrounding words (Lowe and McDonald, 2000; Curran and Moens, 2002a), dependency or case structure (Hindle, 1990; Ruge, 1997; Lin, 1998), and dependency path (Lin and Pantel, 2001; Pado and Lapata, 2007), has been utilized for similarity calculation, and achieved considerable success. However, a major problem which arises when adopting distributional similarity is that it easily yields a huge amount of unique contexts. This can lead to high dimensionality of context space, often up to the order of tens or hundreds of thousands, which makes the calculation computationally impractical. Because not all of the contexts are useful, it is strongly required for the efficiency to eliminate the unwanted contexts to ease the expensive cost.

To tackle this issue, Curran and Moens (2002b) suggest assigning an index vector of *canonical attributes*, i.e., a small number of representative elements extracted from the original vector, to each word. When the comparison is performed, canonical attributes of two target words are firstly consulted, and the original vectors are referred to only if the attributes have a match between them. However, it is not clear whether the condition for canonical attributes they adopted, i.e., that the attributes must be the most weighted subject, direct object, or indirect object, is optimal in terms of the performance.

There are also some existing studies which paid attention to the comparison of context categories for synonym acquisition (Curran and Moens, 2002a; Hagiwara et al., 2006). However, they have conducted only a posteriori comparison based on performance evaluation, and we are afraid that these find-

ings are somewhat limited to their own experimental settings which may not be applicable to completely new settings, e.g., one with a new set of contexts extracted from different sources. Therefore, general quantitative measures which can be used for reduction and selection of any kind of contexts and context categories are strongly required.

Shifting our attention from word similarity to other areas, a great deal of studies on feature selection has been conducted in the literature, especially for text categorization (Yang and Pedersen, 1997) and gene expression classification (Ding and Peng, 2003). Whereas these methods have been successful in reducing feature size while keeping classification performance, the problem of distributional similarity is radically different from that of classification, and whether the same methods are applicable and effective for automatic context selection in the similarity problem is yet to be investigated.

In this paper, we firstly introduce existing quantitative methods for feature selection, namely, DF, TS, MI, IG, CHI2, and show how to apply them to the distributional similarity problem to measure the context importance. We then extracted dependency relations as context from the corpus, and conducted automatic synonym acquisition experiments to evaluate the context selection performance, reducing the unimportant contexts based on the feature selection methods. Finally we extend the context importance to cover context categories (RASP2 grammatical relations), and show that the above methods are also effective in selecting categories.

This paper is organized as follows: in Section 2, five existing context selection methods are introduced, and how to apply classification-based selection methods to distributional similarity is described. In Section 3 and 4, the synonym acquisition method and evaluation measures, AP and CC, employed in the evaluation experiments are detailed. Section 5 includes two main experiments and their results: context reduction and context category selection, along with experimental settings and discussions. Section 6 concludes this paper.

2 Context Selection Methods

In this section, context selection methods proposed for text categorization or information retrieval are introduced. In the following, n and m represent the number of unique words and unique contexts, respectively, and $N(w, c)$ denotes the number of co-occurrence of word w and context c.

2.1 Document Frequency (DF)

Document frequency (DF), commonly used for weighting in information retrieval, is the number of documents a term co-occur with. However, in the distributional similarity settings, DF corresponds to *word frequency*, i.e., the number of unique words the context co-occurs with:

$$df(c) = |\{w|N(w, c) > 0\}|.$$

The motivation of adopting DF as a context selection criterion is the assumption that the contexts shared by many words should be informative. It is to note, however, that the contexts with too high DF are not always useful, since there are some exceptions including so-called *stopwords*.

2.2 Term Strength (TS)

Term strength (TS), proposed by Wilbur and Sirotkin (1992) and applied to text categorization by Yang and Wilbur (1996), measures how likely a term is to appear in "similar documents," and it is shown to achieve a successful outcome in reducing the amount of vocabulary for text retrieval. For distributional similarity, TS is defined as:

$$s(c) = P(c \in C(w_2)|c \in C(w_1)),$$

where (w_1, w_2) is a related word pair and $C(w)$ is a set of contexts co-occurring with the word w, i.e., $C(w) = \{c|N(w, c) > 0\}$. $s(c)$ is calculated, letting P_H be a set of related word pairs, as

$$s(c) = \frac{|\{(w_1, w_2) \in P_H|c \in C(w_1) \cap C(w_2)\}|}{|\{(w_1, w_2) \in P_H|c \in C(w_1)\}|}.$$

What makes TS different from DF is that it requires a training set P_H consisting of related word pairs. We used the test set for class $s = 1$ as P_H described in the next section.

2.3 Formalization of Distributional Similarity

The following methods, MI, IG, and CHI2, are radically different from the above ones, in that they are

designed essentially for "class classification" problems. Thus we formalize distributional similarity as a classification problem as described below.

First of all, we deal with word pairs, instead of words, as the targets of classification, and define features $f_1, ..., f_m$ corresponding to contexts $c_1, ..., c_m$, for each pair. The feature $f_j = 1$ if the two words of the pair has the context c_j in common, and $f_j = 0$ otherwise. Then, we define target class s, so that $s = 1$ when the pair is semantically related, and $s = 0$ if not. These defined, distributional similarity is formalized as a binary classification problem which assigns the word pairs to the class $s \in \{0, 1\}$ based on the features $c_1, ..., c_m$. Finally, to calculate the specific values of the following feature importance measures, we prepare two test sets of related word pairs for class $s = 1$ and unrelated ones for class $s = 0$. This enables us to apply existing feature selection methods designed for classification problems to the automatic context selection.

The two test sets, related and unrelated one, are prepared using the *reference sets* described in Section 4. More specifically, we created 5,000 related word pairs by extracting from synonym pairs in the reference set, and 5,000 unrelated ones by firstly creating random pairs of LDV, whose detail is described later, and then manually making sure that no related pairs are included in these random pairs.

2.4 Mutual Information (MI)

Mutual information (MI), commonly used for word association and co-occurrence weighing in statistical NLP, is the measure of the degree of dependence between two events. The *pointwise* MI value of feature f and class s is calculated as:

$$I(f, s) = \log \frac{P(f, s)}{P(f)P(s)}.$$

To obtain the final context importance, we combine the MI value over both of the classes as $I_{\max}(c_j) = \max_{s \in \{0,1\}} I(f_j, s)$. Note that, here we employed the maximum value of pointwise MI values since it is claimed to be the best in (Yang and Pedersen, 1997), although there can be other combination ways such as weighted average.

2.5 Information Gain (IG)

Information gain (IG), often employed in the machine learning field as a criterion for feature importance, is the amount of gained information of an event by knowing the outcome of the other event, and is calculated as the weighted sum of the pointwise MI values over all the event combinations:

$$G(c_j) = \sum_{f_j \in \{0,1\}} \sum_{s \in \{0,1\}} P(f_j, s) \log \frac{P(f_j, s)}{P(f_j)P(s)}.$$

2.6 χ^2 Statistic (CHI2)

χ^2 statistic (CHI2) estimates the lack of independence between classes and features, which is equal to the summed difference of observed and expected frequency over the contingency table cells. More specifically, letting $F^j_{nm}(n, m \in \{0, 1\})$ be the number of word pairs with $f_j = n$ and $s = m$, and the number of all pairs be N, χ^2 statistic is defined as:

$$\chi^2(c_j) = \frac{N(F_{11}F_{00} - F_{01}F_{10})}{(F_{11} + F_{01})(F_{10} + F_{00})(F_{11} + F_{10})(F_{01} + F_{00})}.$$

3 Synonym Acquisition Method

This section describes the synonym acquisition method, a major and important application of distributional similarity, which we employed for the evaluation of automatic context selection. Here we mention how to extract the original contexts from corpora in detail, as well as the calculation of weight and similarity between words.

3.1 Context Extraction

We adopted dependency structure as the context of words since it is the most widely used and well-performing contextual information in the past studies (Ruge, 1997; Lin, 1998). As the extraction of accurate and comprehensive dependency structure is in itself a difficult task, the sophisticated parser RASP Toolkit 2 (Briscoe et al., 2006) was utilized to extract this kind of word relations. Take the following sentence for example:

> Shipments have been relatively level since January, the Commerce Department noted.

RASP outputs the extracted dependency structure as n-ary relations as follows, which are called *grammatical relations*. Annotations regarding suffix, part of speech tags, offsets for individual words are omitted for simplicity.

```
(ncsubj be Shipment _)
(aux be have)
(xcomp _ be level)
(ncmod _ be relatively)
(ccomp _ level note)
(ncmod _ note since)
(ncsubj note Department _)
(det Department the)
(ncmod _ Department Commerce)
(dobj since January)
```

While the RASP outputs are n-ary relations in general, what we need here is co-occurrences of words and contexts, so we extract the set of co-occurrences of stemmed words and contexts by taking out the target word from the relation and replacing the slot by an asterisk "*":

```
(words)     -  (contexts)
Shipment    -  ncsubj:be:*_
have        -  aux:be:*
be          -  ncsubj:*:Shipment:_
be          -  aux:*:have
be          -  xcomp:_:*:level
be          -  ncmod:_:*:relatively
relatively  -  ncmod:_:be:*
level       -  xcomp:_:be:*
level       -  ccomp:_:*:note
...
```

Summing all these up produces the raw co-occurrence count $N(w, c)$ of word w and context c.

3.2 Similarity Calculation

Although it is possible to use the raw count acquired above for the similarity calculation, directly using the raw count may cause performance degradation, thus we need an appropriate weighting measure. In response to the preliminary experiment results, we employed pointwise mutual information as weight:

$$\mathrm{wgt}(w, c) = \log \frac{P(w, c)}{P(w)P(c)}$$

Here we made a small modification to bind the weight to non-negative such that $\mathrm{wgt}(w, c) \geq 0$, because negative weight values sometimes worsen the performance (Curran and Moens, 2002b). The weighting by PMI is applied *after* the pre-processing including frequency cutoff and context selection.

As for the similarity measure, we used Jaccard coefficient, which is widely adopted to capture overlap proportion of two sets:

$$\frac{\sum_{c \in C(w_1) \cap C(w_2)} \min(\mathrm{wgt}(w_1, c), \mathrm{wgt}(w_2, c))}{\sum_{c \in C(w_1) \cup C(w_2)} \max(\mathrm{wgt}(w_1, c), \mathrm{wgt}(w_2, c))}.$$

4 Evaluation Measures

This section describes the two evaluation methods we employed — average precision (AP) and correlation coefficient (CC).

4.1 Average Precision (AP)

The first evaluation measure, average precision (AP), is a common evaluation scheme for information retrieval, which evaluates how accurately the methods are able to extract synonyms. We first prepare a set of *query words*, for which synonyms are obtained to evaluate the precision. We adopted the Longman Defining Vocabulary (LDV) [1] as the candidate set of query words. For each word in LDV, three existing thesauri are consulted: Roget's Thesaurus (Roget, 1995), Collins COBUILD Thesaurus (Collins, 2002), and WordNet (Fellbaum, 1998). The union of synonyms obtained when the LDV word is looked up as a noun is used as the *reference set*, except for words marked as "idiom," "informal," "slang" and phrases comprised of two or more words. The LDV words for which no noun synonyms are found in any of the reference thesauri are omitted. From the remaining 771 LDV words, 100 query words are randomly extracted, and for each of them the eleven precision values at 0%, 10%, ..., and 100% recall levels are averaged to calculate the final AP value.

4.2 Correlation Coefficient (CC)

The second evaluation measure is correlation coefficient (CC) between the target similarity and the *reference similarity*, i.e., the answer value of similarity for word pairs. The reference similarity is calculated based on the closeness of two words in the tree structure of WordNet. More specifically, the similarity between word w with senses $w_1, ..., w_{m_1}$ and word v with senses $v_1, ..., v_{m_2}$ is obtained as follows. Let the depth of node w_i and v_j be d_i and d_j,

[1] http://www.cs.utexas.edu/users/kbarker/working_notes/ldoce-vocab.html

and the depth of the deepest common ancestors of both nodes be d_{dca}. The similarity is then

$$sim(w, v) = \max_{i,j} sim(w_i, v_j) = \max_{i,j} \frac{2 \cdot d_{\mathrm{dca}}}{d_i + d_j},$$

which takes the value between 0.0 and 1.0. Then, the value of CC is calculated as the correlation coefficient of reference similarities $\boldsymbol{r} = (r_1, r_2, ..., r_n)$ and target similarities $\boldsymbol{s} = (s_1, s_2, ..., s_n)$ over the word pairs in sample set P_s, which is created by choosing the most similar 2,000 word pairs from 4,000 randomly created pairs from LDV. To avoid test-set dependency, all the CC values presented in this paper are the average values of three trials using different test sets.

5 Experiments

Now we describe the experimental settings and the evaluation results of context selection methods.

5.1 Experimental Settings

As for the corpus, New York Times section of English Gigaword [2], consisting of around 914 million words and 1.3 million documents was analyzed to obtain word-context co-occurrences. Frequency cut-off was applied as a pre-processing in order to filter out any words and contexts with low frequency and to reduce computational cost. More specifically, any words w such that $\sum_c \mathrm{tf}(w, c) < \theta_f$ and any contexts c such that $\sum_w \mathrm{tf}(w, c) < \theta_f$, with $\theta_f = 40$, were removed from the co-occurrence data.

Since we set our purpose here to the automatic acquisition of synonymous nouns, only the nouns except for proper nouns were selected. To distinguish nouns, using POS tags annotated by RASP2, any words with POS tags APP, ND, NN, NP, PN, PP were labeled as nouns. This left a total of 40,461 unique words and 139,618 unique context, which corresponds to the number of vectors and the dimensionality of semantic space, respectively.

5.2 Context Reduction

In the first experiment, we show the effectiveness of the five contextual selection methods introduced in Section 2 for context reduction problem. The five measures were calculated for each context, and contexts were sorted by their importance. The change of performance, AP and CC, was calculated on eliminating the low-ranked contexts and varying the proportion of remaining ones, until only 0.2% (279 in number) of the unique contexts are left.

The result is displayed in Figure 1. The overall observation is that the performance not only kept the original level but also slightly improved even during the "aggressive" reduction when more than 80% of the original contexts were eliminated and less than 20,000 contexts were left. It was not until 90% (approx. 10,000 remaining) elimination that the AP values began to fall. The tendency of performance change was almost the same for AP and CC, but we observe a slight difference regarding which of the five measures were effective. More specifically, TS, IG and CHI2 worked well for AP, and DF, TS, while CHI2 did for CC. On the whole, TS and CHI2 were performing the best, whereas the performance of MI quickly worsened. Although the task is different, this experiment showed a very consistent result compared with the one of Yang and Pedersen's (1997). This means that feature selection methods are also effective for context selection in distributional similarity, and our formalization of the problem described in Section 2 turned out to be appropriate for the purpose.

5.3 Context Category Selection

We are then naturally interested in what kinds of contexts are included in these top-ranked effective ones and how much they affect the overall performance. To investigate this, we firstly built a set of *elite contexts*, by gathering each top 10% (13,961 in number) contexts chosen by DF, TS, IG, and CHI2, and obtaining the intersection of these four top-ranked contexts. It was found that these four had a great deal of overlap among them, the number of which turned out to be 6,440.

Secondly, to measure the degree of effect a context category has, we defined *category importance* as the sum of all IG values of the contexts which belong to the category. The reason is that, (a) IG was one of the best-performing criteria as the previous experiment showed, and (b) IG value for a set of contexts can be calculated as the sum of IG values of individual elements, assuming that all the contexts

[2] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

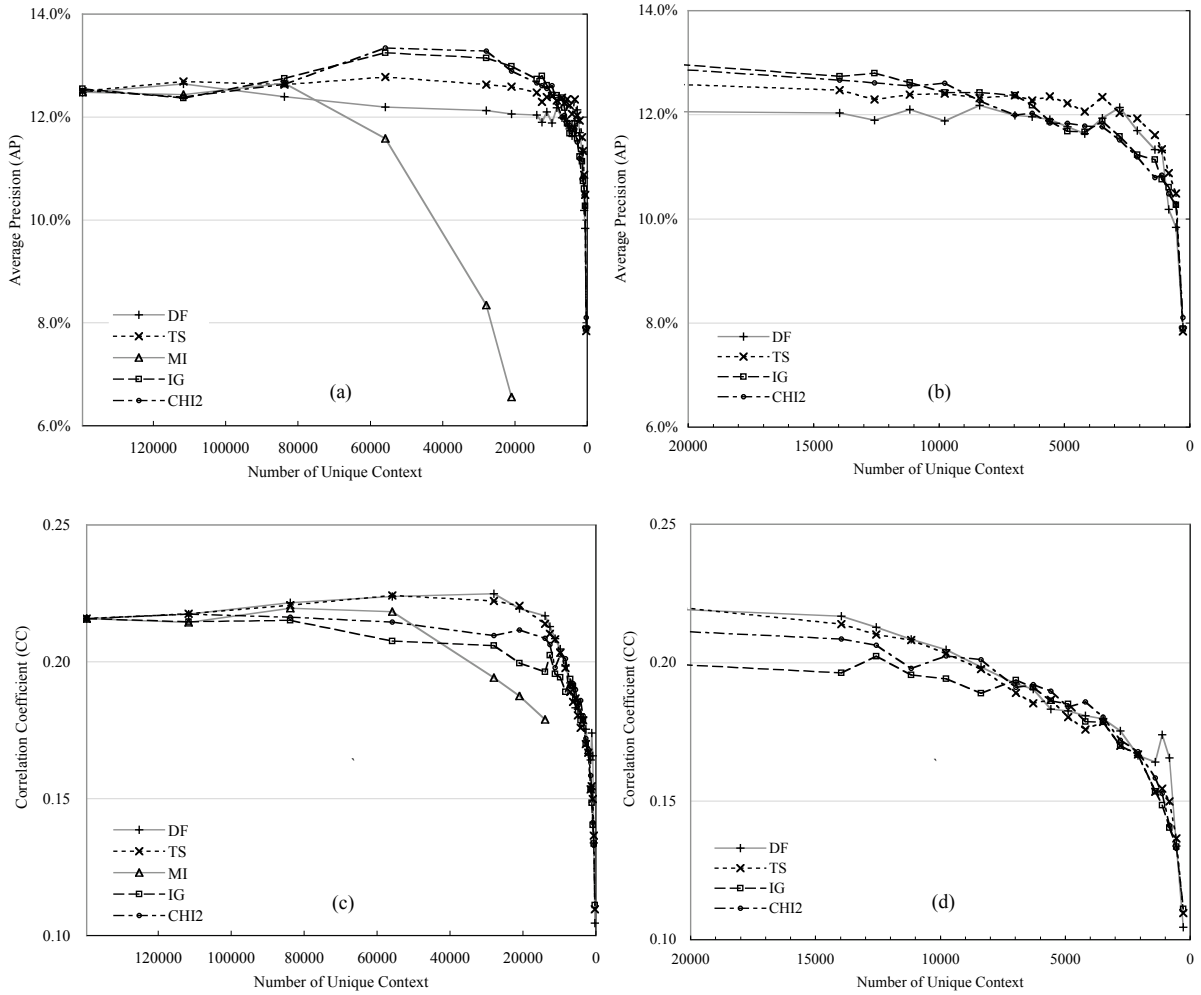


Figure 1: Performance of synonym acquisition on automatic context reduction
(a) The overall view and (b) the close-up of 0 to 20,000 unique contexts for AP,
and (c) the overall view and (b) the close-up for CC

are mutually independent, which is a naive but practical assumption because of the high independence of acquired contexts from corpora.

For the categories: `ncsubj`, `dobj`, `obj`, `obj2`, `ncmod`, `xmod`, `cmod`, `ccomp`, `det`, `ta`, based on the RASP2 grammatical relations which occur frequently (more than 1.0%) in the corpus, their category importance within the elite context set was computed and showed in Figure 2. The graph also shows the performance of individual context categories, calculated when each category was separately extracted from the entire corpus. The result indicates that there is a considerable correlation ($r = 0.760$) between category importance and performance, which means it is possible to predict the final performance of any context categories by calculating their category importance values in the limited size of selected context set.

As for the qualitative difference of category types, the result also shows the effectiveness of modification (`ncmod`) category, which is consistent with the result (Hagiwara et al., 2006) that `mod` is more contributing than `subj` and `obj`, which have been extensively used in the past. However, it can be seen that the reason why the `ncmod` performs well may be only because it is the largest category in size (2,515

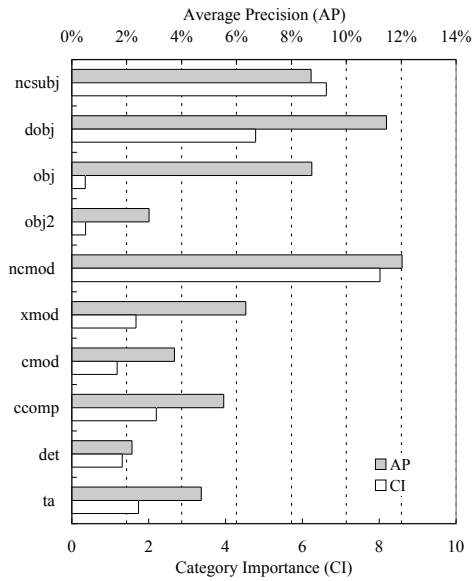


Figure 2: Performance of synonym acquisition vs context category importance

in the elite contexts). The investigation of the relations between context size and performance should be conducted in the future.

6 Conclusion

In this study, we firstly introduced feature selection methods, previously proposed for text categorization, and showed how to apply them for automatic context selection for distributional similarity by formalizing the similarity problem as classification. We then extracted dependency-based context from the corpus, and conducted evaluation experiments on automatic synonym acquisition.

The experimental results showed that while keeping or even improving the original performance, it is possible to eliminate a large proportion of contexts (almost up to 90%). We also extended the context importance to cover context categories based on RASP2 grammatical relations, and showed a considerable correlation between the importance and the actual performance, suggesting the possibility of automatic context category selection.

As the future works, we should further discuss other kinds of formalization of distributional similarity and their impact, because we introduced and only briefly described a quite simple formalization model in Section 2.3. More detailed investigations on the contributions of sub-categories of contexts, and other contexts than dependency structure, such as surrounding words and dependency path, is also the future work.

References

Ted Briscoe, John Carroll and Rebecca Watson. 2006. The Second Release of the RASP System. *Proc. of the COLING/ACL 2006 Interactive Presentation Sessions*, 77–80.

Collins. 2002. Collins Cobuild Major New Edition CD-ROM. HarperCollins Publishers.

James R. Curran and Marc Moens. 2002. Scaling Context Space. *Proc. of ACL 2002*, 231–238.

James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In Workshop on Unsupervised Lexical Acquisition. *Proc. of ACL SIGLEX*, 231–238.

Chris Ding and Hanchuan Peng. 2003. Minimum Redundancy Feature Selection from Microarray Gene Expression Data. *Proc. of the IEEE Computer Society Conference on Bioinformatics*, 523–528.

Editors of the American Heritage Dictionary. 1995. Roget's II: The New Thesaurus, 3rd ed. *Houghton Mifflin*.

Christiane Fellbaum. 1998. *WordNet: an electronic lexical database*, MIT Press.

Masato Hagiwara, Yasuhiro Ogawa, Katsuhiko Toyama. 2006. Selection of Effective Contextual Information for Automatic Synonym Acquisition. *Proc. of COLING/ACL 2006*, 353–360.

Zellig Harris. 1985. Distributional Structure. Jerrold J. Katz (ed.) *The Philosophy of Linguistics*. Oxford University Press. 26–47

Donald Hindle. 1990. Noun classification from predicate-argument structures. *Proc. of the 28th Annual Meeting of the ACL*, 268–275.

Will Lowe and Scott McDonald. 2000. The direct route: Mediated priming in semantic space. *Proc. of the 22nd Annual Conference of the Cognitive Science Society*, 675–680.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. *Proc. of COLING/ACL 1998*, 786–774.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, Volume 7, Issue 4, 343–360.

Seastian Pado and Mirella Lapata. 2007. Dependency-Based Construction of Semantic Space Models *Computational Linguistics*, Volume 33, Issue 2, 161–199.

Gerda Ruge. 1997. Automatic detection of thesaurus relations for information retrieval applications. *Foundations of Computer Science: Potential - Theory - Cognition*, LNCS, Volume 1337, 499–506, Springer Verlag, Berlin, Germany.

Yiming Yang and John Wilbur. 1996. Using corpus statistics to remove redundant words in text categorization. *Journal of the American Society for Information Science*, Volume 47, Issue 5, 357–369.

Yiming Yang and Jan O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. *Proc. of ICML 97*, 412–420.

John Wilbur and Karl Sirotkin. 1992. The automatic identification of stop words. *Journal of Information Science*, 45–55.

Gloss-Based Semantic Similarity Metrics for Predominant Sense Acquisition

Ryu Iida
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
ryu-i@is.naist.jp

Diana McCarthy and **Rob Koeling**
University of Sussex
Falmer, East Sussex
BN1 9QH, UK
{dianam,robk}@sussex.ac.uk

Abstract

In recent years there have been various approaches aimed at automatic acquisition of predominant senses of words. This information can be exploited as a powerful back-off strategy for word sense disambiguation given the zipfian distribution of word senses. Approaches which do not require manually sense-tagged data have been proposed for English exploiting lexical resources available, notably WordNet. In these approaches distributional similarity is coupled with a semantic similarity measure which ties the distributionally related words to the sense inventory. The semantic similarity measures that have been used have all taken advantage of the hierarchical information in WordNet. We investigate the applicability to Japanese and demonstrate the feasibility of a measure which uses only information in the dictionary definitions, in contrast with previous work on English which uses hierarchical information in addition to dictionary definitions. We extend the definition based semantic similarity measure with distributional similarity applied to the words in different definitions. This increases the recall of our method and in some cases, precision as well.

1 Introduction

Word sense disambiguation (WSD) has been an active area of research over the last decade because many researches believe it will be important for applications which require, or would benefit from, some degree of semantic interpretation. There has been considerable skepticism over whether WSD will actually improve performance of applications, but we are now starting to see improvement in performance due to WSD in cross-lingual information retrieval (Clough and Stevenson, 2004; Vossen et al., 2006) and machine translation (Carpuat and Wu, 2007; Chan et al., 2007) and we hope that other applications such as question-answering, text simplification and summarisation might also benefit as WSD methods improve.

In addition to contextual evidence, most WSD systems exploit information on the most likely meaning of a word regardless of context. This is a powerful back-off strategy given the skewed nature of word sense distributions. For example, in the English coarse grained all words task (Navigli et al., 2007) at the recent SemEval Workshop the baseline of choosing the most frequent sense using the first WordNet sense attained precision and recall of 78.9% which is only a few percent lower than the top scoring system which obtained 82.5%. This finding is in line with previous results (Snyder and Palmer, 2004). Systems using a first sense heuristic have relied on sense-tagged data or lexicographer judgment as to which is the predominant sense of a word. However sense-tagged data is expensive and furthermore the predominant sense of a word will vary depending on the domain (Koeling et al., 2005; Chan and Ng, 2007).

One direction of research following McCarthy et al. (2004) has been to learn the most predominant

sense of a word automatically. McCarthy et al's method relies on two methods of similarity. Firstly, distributional similarity is used to estimate the predominance of a sense from the number of distributionally similar words and the strength of their distributional similarity to the target word. This is done on the premise that more prevalent meanings have more evidence in the corpus data used for the distributional similarity calculations and the distributionally similar words (nearest neighbours) to a target reflect the more predominant meanings as a consequence. Secondly, the senses in the sense inventory are linked to the nearest neighbours using semantic similarity which incorporates information from the sense inventory. It is this semantic similarity measure which is the focus of our paper in the context of the method for acquiring predominant senses.

Whilst the McCarthy et al.'s method works well for English, other inventories do not always have WordNet style resources to tie the nearest neighbours to the sense inventory. WordNet has many semantic relations as well as glosses associated with its synsets (near synonym sets). While traditional dictionaries do not organise senses into synsets, they do typically have sense definitions associated with the senses. McCarthy et al. (2004) suggest that dictionary definitions can be used with their method, however in the implementation of the measure based on dictionary definitions that they use, the dictionary definitions are extended to those of related words using the hierarchical structure of WordNet (Banerjee and Pedersen, 2002). This extension to the original method (Lesk, 1986) was proposed because there is not always sufficient overlap of the individual words for which semantic similarity is being computed. In this paper we refer to the original method (Lesk, 1986) as **lesk** and the extended measure proposed by Banerjee and Pedersen as **Elesk**.

This paper investigates the potential of using the overlap of dictionary definitions with the McCarthy et al.'s method. We test the method for obtaining a first sense heuristic using two publicly available datasets of sense-tagged data in Japanese, EDR (NICT, 2002) and the SENSEVAL-2 Japanese dictionary task (Shirai, 2001). We contrast an implementation of **lesk** (Lesk, 1986) which uses only dictionary definitions with the Jiang-Conrath measure (**jcn**) (Jiang and Conrath, 1997) which uses manually produced hyponym links and was used previously for this purpose on English datasets (McCarthy et al., 2004). The **jcn** measure is only applicable to the EDR dataset because the dictionary has hyponymy links which are not available in the SENSEVAL-2 Japanese dictionary task. We also propose a new extension to **lesk** which does not require hand-crafted hyponym links but instead uses distributional similarity to increase the possibilities for overlap of the word definitions. We refer to this new measure as **DSlesk**. We compare this to the original **lesk** on both datasets and show that it increases recall, and sometimes precision too whilst not requiring hyponym links.

In the next section we place our contribution in relation to previous work. In section 3 we summarise the methods we adopt from previous work, and describe our proposal for a semantic similarity method that can supplement the information from dictionary definitions with information from raw text. In section 4 we describe the experiments on EDR and the SENSEVAL-2 Japanese dictionary task and we conclude in section 5.

2 Related Work

This work builds upon that of McCarthy et al. (2004) which acquires predominant senses for target words from a large sample of text using distributional similarity (Lin, 1998) to provide evidence for predominance. The evidence from the distributional similarity is allocated to the senses using semantic similarity from WordNet (Patwardhan and Pedersen, 2003). We will describe the method more fully below in section 3. McCarthy et al. (2004) reported results for English using their automatically acquired first sense heuristic on SemCor (Miller et al., 1993) and the SENSEVAL-2 English all words dataset (Snyder and Palmer, 2004). The results from this are promising, given that hand-labelled data is not required. On polysemous nouns from SemCor they obtained 48% WSD using their method with **Elesk** and 46% with **jcn** where the random baseline was 24% and the upper-bound was 67% (derived from the SemCor test data itself). On SENSEVAL-2 all words dataset using the **jcn** measure [1] they obtained 63% recall which is encouraging compared to the

[1] They did not apply **lesk** to this dataset.

SemCor heuristic which obtained 68% but requires hand-labelled data. The upper-bound on the dataset was 72% from the test data itself. These results crucially depend on the information in the sense inventory WordNet. WordNet contains hierarchical relations between word senses which are used in both **jcn** and **Elesk**. There is an issue that such information may not be available in other sense inventories, and other inventories will be needed for other languages. In this paper, we implement the **lesk** semantic similarity (Lesk, 1986) for the two Japanese lexicons used in our test datasets, i) the EDR dictionary (NICT, 2002) ii) the Iwanami Kokugo Jiten Dictionary (Nishio et al., 1994). We investigate the potential of **lesk** and **jcn**, where the latter is applicable. In addition to implementing the original **lesk** measure, we propose an extension to the method inspired by Mihalcea et al. (2006). Mihalcea et al. (2006) used various text based similarity measures, including WordNet and corpus based similarity methods, to determine if two phrases are paraphrases. They contrasted this approach with previous methods which used overlap of the words between the candidate paraphrases. For each word in each of the two texts they obtain the maximum similarity between the word and any of the words from the putative paraphrase. The similarity scores for each word of both phrases contribute to an overall semantic similarity between 0 and 1 and a threshold of 0.5 is used to decide if the candidate phrases are paraphrases. In our work, we compare glosses of words senses (senses of the target word and senses of the nearest neighbour) rather than paraphrases. In this approach we extend the definition overlap by considering the distributional similarity (Lin, 1998) rather than identify of the words in the two definitions.

In addition to McCarthy et al. (2004) there are other approaches to finding predominant senses. Chan and Ng (2005) use parallel data to provide estimates for sense frequency distributions to feed into a supervised WSD system. Mohammad and Hirst (2006) propose an approach to acquiring predominant senses from corpora which makes use of the category information in the Macquarie Thesaurus (Barnard, 1986). Lexical chains (Galley and McKeown, 2003) may also provide a useful first sense heuristic (Brody et al., 2006) but are produced using WordNet relations. We use the McCarthy et al. approach because this is applicable without aligned corpus data, semantic category and relation information and is applicable to any language assuming the minimum requirements of i) dictionary definitions associated with the sense inventory and ii) raw corpus data. We adapt their technique to remove the reliance on hyponym links.

3 Gloss-based semantic similarity

We first summarise the McCarthy et al. method and the WordNet based semantic similarity functions (**jcn** and **Elesk**) that they use for automatic acquisition of a first sense heuristic applied to disambiguation of English WordNet datasets. We then describe the additional semantic similarity method that we propose for comparison with **lesk** and **jcn**.

McCarthy et al. use a distributional similarity thesaurus acquired from corpus data using the method of Lin (1998) for finding the predominant sense of a word where the senses are defined by WordNet. The thesaurus provides the k nearest neighbours to each target word, along with the distributional similarity score between the target word and its neighbour. The WordNet similarity package (Patwardhan and Pedersen, 2003) is used to weight the contribution that each neighbour makes to the various senses of the target word.

Let w be a target word and $N_w = \{n_1, n_2 ... n_k\}$ be the ordered set of the top scoring k neighbours of w from the thesaurus with associated distributional similarity scores $\{dss(w,n_1), dss(w,n_2), ... dss(w,n_k)\}$ using (Lin, 1998). Let $senses(w)$ be the set of senses of w for each sense of w ($ws_i \in senses(w)$) a ranking is obtained using:

$$Prevalence\ Score(ws_i) = \sum_{n_j \in N_w} dss(w,n_j) \times \frac{wnss(ws_i, n_j)}{\sum_{ws_{i'} \in senses(w)} wnss(ws_{i'}, n_j)} \quad (1)$$

where $wnss$ is the maximum WordNet similarity score between ws_i and the WordNet sense of the neighbour (n_j) that maximises this score. McCarthy et al. compare two different WordNet similarity scores, **jcn** and **Elesk**.

jcn (Jiang and Conrath, 1997) uses corpus data to estimate a frequency distribution over the classes

(synsets) in the WordNet hierarchy. Each synset, is incremented with the frequency counts from the corpus of all words belonging to that synset, directly or via the hyponymy relation. The frequency data is used to calculate the "information content" (IC) of a class or sense (s):

$$IC(s) = -log(p(s))$$

Jiang and Conrath specify a distance measure between two senses ($s1, s2$):

$$D_{jcn}(s1, s2) = IC(s1) + IC(s2) - 2 \times IC(s3)$$

where the third class ($s3$) is the most informative, or most specific, superordinate synset of the two senses $s1$ and $s2$. This is transformed from a distance measure in the WordNet Similarity package by taking the reciprocal:

$$jcn(s1, s2) = 1/D_{jcn}(s1, s2)$$

McCarthy et al. use the above measure with ws_i as $s1$ and whichever sense of the neigbour (n_j) that maximises this WordNet similarity score.

Elesk (Banerjee and Pedersen, 2002) extends the original **lesk** algorithm (Lesk, 1986) so we describe that original algorithm **lesk** first. This simply calculates the overlap of the content words in the definitions, frequently referred to as glosses, of the two word senses.

$$lesk(s1, s2) = \sum_{a \in g_1} member(a, g_2)$$

$$member(a, g_2) = \begin{cases} 1 & \text{if } a \text{ appears in } g_2 \\ 0 & otherwise \end{cases}$$

where g_1 is the gloss of word sense $s1$, g_2 is the gloss of $s2$ and a is one of words appearing in g_1. In **Elesk** which McCarthy et al. use the measure is extended by considering related synsets to $s1$ and $s2$, again where $s1$ is ws_i and $s2$ is the sense from all senses of n_j that maximises the **Elesk** WordNet similarity score. **Elesk** relies heavily on the relationships that are encoded in WordNet such as hyponymy and meronymy. Not all languages have resources supplied with these relations, and where they are supplied there may not be as much detail as there is in WordNet.

In this paper we will examine the use of **jcn** and the original **lesk** in Japanese on the EDR dataset to see how well the pure definition based measure fares compared to one using hyponym links. EDR has hyponym links so we can make this comparison. The performance of **jcn** will depend on the coverage of the hyponym links. For **lesk** meanwhile there is an issue that using only overlap of sense definitions may give poor results because the sense definitions are usually succinct and the overlap of words may be low. For example, given the glosses for the words *pigeon* and *bird*:[2]

> *pigeon: a fat grey and white bird with short legs.*
> *bird: a creature that is covered with feathers and has wings and two legs.*

If only content words are considered then there is only one word (*leg*) which overlaps in the two glosses, so the resultant **lesk** score is low (1) even though the word *pigeon* is intuitively similar to *bird*.

The **Elesk** extension addressed this issue using WordNet relations to extend the definitions over which the overlap is calculated for a given pair of senses. We propose addressing the same issue using corpus data to supplement the **lesk** overlap measure. We propose using distributional similarity (using (Lin, 1998)) as an approximation of semantic distance between the words in the two glosses, rather than requiring an exact match. We refer to this measure as **DSlesk** as defined:

$$DSlesk(s1, s2) = \frac{1}{|a \in g_1|} \sum_{a \in g_1} \max_{b \in g_2} dss(a, b) \quad (2)$$

where g_1 is the gloss of word sense $s1$, g_2 is the gloss of $s2$, again $s1$ is the target word sense ws_i in equation 1 for which we are obtaining the predominance ranking score and $s2$ is whichever sense of the neighbour (n_j) in equation 1 which maximises this semantic similarity score, as McCarthy et al. did with the *wnss* in equation 1. a (b) is a word appearing in g_1 (g_2).

In the calculation of equation (2), we first extract the most similar word b from g_2 to each word (a) in

[2]These two glosses are defined in OXFORD Advanced Learner's Dictionary.

$dss(bird, creature) = 0.84$,	$dss(bird, feather) = 0.77$,
$dss(bird, wing) = 0.55$,	$dss(bird, leg) = 0.43$,
$dss(leg, creature) = 0.56$,	$dss(leg, feather) = 0.66$,
$dss(leg, wing) = 0.74$,	$dss(leg, leg) = 1.00$

Figure 1: Examples of distributional similarity

the gloss of $s1$. We then output the average of the maximum distributional similarity of all the words in g_1 to any of the words in g_2 as the similarity score between $s1$ and $s2$. We acknowledge that **DSlesk** is not symmetrical since it depends on the number of words in the gloss of $s1$, but not $s2$. Also our summation is over these words in $s1$ and we are not looking for identity but maximum distributional similarity with any of the words in g_2 so the summation will not give the same result as if we did the summation over the words in g_2. It is perfectly reasonable to have a semantic similarity measure which is not symmetrical. One may want a measure where a more specific sense, such as the meat sense of *chicken* is closer to the "animal flesh used as food" sense of meat than vice versa. We do not believe that this asymmetry is problematic for our application as all the senses of w which we are ranking are all treated equally with respect to the neighbour n, and the ranking measure is concerned with finding evidence for the meaning of w, which we do by focusing on its definitions, and not the meaning of n. It would however be worthwhile investigating symmetrical versions of the score in the future.

Here is an example given the definitions of *bird* and *pigeon* above and the distributional similarity scores of all combinations of the two nouns as shown in Figure 1. In this case, the similarity is estimated as $1/2(0.84 + 1.00) = 0.92$.

4 Experiments

To investigate how well the McCarthy et al. method ports to other language, we conduct empirical evaluation of word sense disambiguation by using the two available sense-tagged datasets, EDR and the SENSEVAL-2 Japanese dictionary task. In the experiments, we compare the three semantic similarities, **jcn**, **lesk** and **DSlesk**[3], for use in the method to find the most likely sense in the set of word senses defined in each inventory following the approach of McCarthy et al. (2004). For the thesaurus construction we used <verb, case, noun> triplets extracted from Japanese newspaper articles (9 years of the Mainichi Shinbun (1991-1999) and 10 years of the Nihon Keizai Shinbun (1991-2000)) and parsed by CaboCha (Kudo and Matsumoto, 2002). This resulted in 53 million triplet instances for acquiring the distributional thesaurus. We adopt the similarity score proposed by Lin (1998) as the distributional similarity score and use 50 nearest neighbours in line with McCarthy et al.

For the random baseline we select one word sense at random for each word token and average the precision over 100 trials. For contrast with a supervised approach we show the performance if we use hand-labelled training data for obtaining the predominant sense of the test words. This method usually outperforms an automatic approach, but crucially relies on there being hand-labelled data which is expensive to produce. The method cannot be applied where there is no hand-labelled training data, it will be unreliable for low frequency data and a general dataset may not be applicable when one moves to domain specific text (Koeling et al., 2005). Since we are not using context for disambiguation, but just a first sense heuristic, we also give the upper-bound which is the first sense heuristic calculated from the test data itself.

4.1 EDR

We conduct empirical evaluation using 3,836 polysemous nouns in the sense-tagged corpus provided with EDR (183,502 instances) where the glosses are defined in the EDR dictionary. We evaluated on this dataset using WSD precision and recall of this corpus using only our first-sense heuristic (no context). The results are shown in Table 1. The WSD performance of all the automatic methods is much lower than the supervised method, however, the main point of this paper is to compare the McCarthy et al. method for finding a first sense in Japanese using **jcn**, **lesk** and

[3] **Elesk** can be used when several semantic relations such as hypnoymy and meronomy are available. However, we cannot directly apply **Elesk** as it was used in (McCarthy et al., 2004) to our experiments because the meronomy relation is not defined in the EDR dictionary. In the experiments reported here we focus on the comparison of the three similarity measures **jcn**, **lesk** and **DSlesk** for use in the method to determine the predominant sense of each word. We leave further exploration of other adaptations of semantic similarity scores for future work.

Table 1: Results of EDR

	recall	precision
baseline	0.402	0.402
jcn	0.495	0.495
lesk	0.474	0.488
DSlesk	0.495	0.495
upper-bound	0.745	0.745
supervised	0.731	0.731

Table 2: Precision on EDR at low frequencies

	all	freq ≤ 10	freq ≤ 5
baseline	0.402	0.405	0.402
jcn	0.495	0.445	0.431
lesk	0.474	0.448	0.426
DSlesk	0.495	0.453	0.433
upper-bound	0.745	0.674	0.639
supervised	0.731	0.519	0.367

DSlesk. Table 1 shows that **DSlesk** is comparable to **jcn** without the requirement for semantic relations such as hyponymy.

Furthermore, we evaluate precision of each method at low frequencies of words (≤ 10, ≤ 5), shown in Table 2. Table 2 shows that all methods for finding a predominant sense outperform the supervised one for items with little data (≤ 5), indicating that these methods robustly work even for low frequency data where hand-tagged data is unreliable.

Whilst the results are significantly different to the baseline [4] we note that the difference to the random baseline is less than for McCarthy et al. who obtained 48% for **Elesk** on polysemous nouns in SemCor and 46% for **jcn** against a random baseline of 24%. These differences are probably explained by differences in the lexical resources. Both **Elesk** and **jcn** rely on semantic relations including hyponymy with **Elesk** also using the glosses. **jcn** in both approaches use the hyponym links. WordNet 1.6 (used by McCarthy et al.) has 66025 synsets with 66910 hyponym links between these [5]. For EDR there are 166868 nodes (word sense groupings) and 53747 hyponym links. So in EDR the ratio of these links to the nodes is much lower. This and other differences between EDR and WordNet are likely to be the reason for the difference in results.

[4] For significance testing we used McNemar's test $\alpha = 0.05$.

[5] These figures are taken from http://www.lsi.upc.es/~batalla/wnstats.html#wn16

Table 3: Results of SENSEVAL-2

	precision = recall	
	fine	coarse
baseline	0.282	0.399
lesk	0.344	0.501
DSlesk	0.386	0.593
upper-bound	0.747	0.834
supervised	0.742	0.842

4.2 SENSEVAL-2

We also evaluate the performance using the Japanese dictionary task in SENSEVAL-2 (Shirai, 2001). In this experiment, we use 50 nouns (5,000 instances). For this task, since semantic relations such as hyponym links are not defined, use of **jcn** is not possible. Therefore, we just compare **lesk** and **DSlesk** along with our random baseline, the supervised approach and the upper-bound as before.

The results are evaluated in two ways; one is for fine-grained senses in the original task definition and the other is coarse-grained version which is evaluated discarding the finer categorical information of each definition. The results are shown in Table 3. As with the EDR results, all unsupervised methods significantly outperform the baseline method, though the supervised methods still outperform the unsupervised ones. In this experiment, **DSlesk** is also significantly better than **lesk** in both fine and coarse-grained evaluations. It indicates that applying distributional similarity score to calculating inter-gloss similarities improves performance.

5 Conclusion

In this paper, we examined different measures of semantic similarity for finding a first sense heuristic for WSD automatically in Japanese. We defined a new gloss-based similarity (**DSlesk**) and evaluated the performance on two Japanese WSD datasets, outperforming **lesk** and achieving a performance comparable to the **jcn** method which relies on hyponym links which are not always available.

There are several issues for future directions of automatic detection of a first sense heuristic. In this paper, we proposed an adaptation of the **lesk** measure of gloss-based similarity, by using the average similarity between nouns in the two glosses under comparison in a bag-of-words approach without recourse to other information. However, it would be worthwhile exploring other information in the glosses, such as words of other PoS and predicate argument relations. We also hope to investigate applying alignment techniques introduced for entailment recognition (Hickl and Bensley, 2007).

Another important issue in WSD is to group fine-grained word senses into clusters, making the task suitable for NLP applications (Ide and Wilks, 2006). We believe that our gloss-based similarity **DSlesk** might be very suitable for this task and we plan to investigate the possibility.

There are other approaches we would like to explore in future. Mihalcea (2005) uses dictionary definitions alongside graphical algorithms for unsupervised WSD. Whilst the results are not directly comparable to ours because we have not included contextual evidence in our models, it would be worthwhile exploring if unsupervised graphical models using only the definitions we have in our lexical resources can perform WSD on a document and give more reliable first sense heuristics.

Acknowledgements

This work was supported by the UK EPSRC project EP/C537262 'Ranking Word Senses for Disambiguation: Models and Applications', and a UK Royal Society Dorothy Hodgkin Fellowship to the second author. We would like to thank John Carroll for several useful discussions on this work.

References

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-02)*, Mexico City.

J.R.L. Barnard, editor. 1986. *Macquaire Thesaurus*. Macquaire Library, Sydney.

Samuel Brody, Roberto Navigli, and Mirella Lapata. 2006. Ensemble methods for unsupervised wsd. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, July. Association for Computational Linguistics.

Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, pages 61–72, Prague, Czech Republic, June. Association for Computational Linguistics.

Yee Seng Chan and Hwee Tou Ng. 2005. Word sense disambiguation with distribution estimation. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, Edinburgh, Scotland.

Yee Seng Chan and Hwee Tou Ng. 2007. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June. Association for Computational Linguistics.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June. Association for Computational Linguistics.

Paul Clough and Mark Stevenson. 2004. Evaluating the contribution of EuroWordNet and word sense disambiguation to cross-language retrieval. In *Second International Global WordNet Conference (GWC-2004)*, pages 97–105.

Michel Galley and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 1486–1488. Morgan Kaufmann.

Andrew Hickl and Jeremy Bensley. 2007. A discourse commitment-based framework for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171–176.

Nancy Ide and Yorick Wilks. 2006. Making sense about sense. In Eneko Agirre and Phil Edmonds, editors, *Word Sense Disambiguation, Algorithms and Applications*, pages 47–73. Springer.

Jay Jiang and David Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research in Computational Linguistics*, Taiwan.

Rob Koeling, Diana McCarthy, and John Carroll. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the joint conference on Human Language Technology and Empirical methods in Natural Language Processing*, pages 419–426, Vancouver, B.C., Canada.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning 2002* (*CoNLL*), pages 63–69.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from and ice cream cone. In *Proceedings of the ACM SIGDOC Conference*, pages 24–26, Toronto, Canada.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL 98*, Montreal, Canada.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 280–287, Barcelona, Spain.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the American Association for Artificial Intelligence (AAAI 2006)*, Boston, MA, July.

Rada Mihalcea. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the joint conference on Human Language Technology and Empirical methods in Natural Language Processing*, Vancouver, B.C., Canada.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 303–.308. Morgan Kaufman.

Saif Mohammad and Graeme Hirst. 2006. Determining word sense dominance using a thesaurus. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 121–128, Trento, Italy, April.

Roberto Navigli, C. Litkowski, Kenneth, and Orin Hargraves. 2007. SemEval-2007 task 7: Coarse-grained English all-words task. In *Proceedings of ACL/SIGLEX SemEval-2007*, pages 30–35, Prague, Czech Republic.

NICT. 2002. EDR electronic dictionary version 2.0, technical guide. http://www2.nict.go.jp/kk/e416/EDR/.

Minoru Nishio, Etsutaro Iwabuchi, and Shizuo Mitzutani. 1994. Iwanami kokugo jiten dai go han.

Siddharth Patwardhan and Ted Pedersen. 2003. The CPAN WordNet::Similarity Package. http://search.cpan.org/author/SID/WordNet-Similarity-0.03/.

Kiyoaki Shirai. 2001. SENSEVAL-2 Japanese Dictionary Task. In *Proceedings of the* SENSEVAL-2 *workshop*, pages 33–36.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of the ACL* SENSEVAL-3 *workshop*, pages 41–43, Barcelona, Spain.

Piek Vossen, German Rigau, Inaki Alegria, Eneko Agirre, David Farwell, and Manuel Fuentes. 2006. Meaningful results for information retrieval in the meaning project. In *Proceedings of the 3rd Global WordNet Conference*. http://nlpweb.kaist.ac.kr/gwc/.

Benchmarking Noun Compound Interpretation

Su Nam Kim and Timothy Baldwin
Department of Computer Science and Software Engineering
and
NICTA Victoria Lab
University of Melbourne, VIC 3010 Australia
{snkim,tim}@csse.unimelb.edu.au

Abstract

In this paper we provide benchmark results for two classes of methods used in interpreting noun compounds (NCs): semantic similarity-based methods and their hybrids. We evaluate the methods using 7-way and binary class data from the nominal pair interpretation task of SEMEVAL-2007.[1] We summarize and analyse our results, with the intention of providing a framework for benchmarking future research in this area.

1 Introduction

This paper reviews a range of simple and hybrid approaches to noun compound (NC) interpretation. The interpretation of NCs such as *computer science* and *paper submission* involves predicting the **semantic relation** (SR) that underlies a given NC. For example, *student price* conventionally expresses the meaning that a *student* benefits from the *price* (SR = BENEFICIARY), while *student protest* conventionally means a *student* undertaking a *protest* (SR = AGENT).[2]

NCs are formed from simplex nouns with high productivity. The huge number of possible NCs and potentially large number of SRs makes NC interpretation a very difficult problem. In the past, much NC interpretation work has been carried out which targets particular NLP applications such as information extraction, question-answering and machine translation. Unfortunately, much of it has not gained traction in real-world applications as the accuracy of the methods has not been sufficiently high over open-domain data. Most prior work has been carried out under specific assumptions and with one-off datasets, which makes it hard to analyze performance and to build hybrid methods. Additionally, disagreement in the inventory of SRs and a lack of resource sharing has hampered comparative evaluation of different methods.

The first step in NC interpretation is to define a set of SRs. Levi (1979), for example, proposed a system of 9 SRs, while others have proposed classifications with 20-30 SRs (Finin, 1980; Barker and Szpakowicz, 1998; Moldovan et al., 2004). Smaller sets tend to have reduced coverage due to coarse granularity, whereas larger sets tend to be too fine grained and suffer from low inter-annotator agreement. Additionally pragmatic/contextual differentiation leads to difficulties in defining and interpreting SRs (Downing, 1977; SparckJones, 1983).

Recent attempts in the area of NC interpretation have taken two basic approaches: analogy-base interpretation (Rosario, 2001; Moldovan et al., 2004; Kim and Baldwin, 2005; Girju, 2007) and semantic disambiguation relative to an underlying predicate or semantically-unambiguous paraphrase (Vanderwende, 1994; Lapata, 2002; Kim and Baldwin, 2006; Nakov, 2006). Most methods employ rich ontologies and ignore the context of use, supporting the claim by Fan (2003) that axioms and ontological distinctions are more important than detailed knowledge of specific nouns for NC interpretation. Additionally, most approaches use supervised learning, raising questions about the generality of the test and

[1] The 4th International Workshop on Semantic Evaluation

[2] SRs used in the examples are taken from Barker and Szpakowicz (1998).

training data sets and the effectiveness of the algorithms in different domains (coverage of SRs over the NCs is another issue).

Our aim in this paper is to compare and analyze existing NC interpretation methods over a common, publicly available dataset. While recent research has made significant progress, bringing us one step closer to practical applicability in NLP applications, no direct comparison or analysis of the approaches has been attempted to date. As a result, it is hard to determine which approach is appropriate in a given domain or build hybrid methods based on prior approaches. We also investigate the impact on performance of relaxing assumptions made in the original research, to compare different approaches in an identical setting.

In the remainder of the paper, we review the research background and NC interpretation methods in Section 2, describe the methods and system architectures in Section 3, detail the datasets used in our experiments in Section 4, carry out a system evaluation in Section 5 and Section 6, and finally present a discussion and conclusions in Section 7 and Section 8, respectively.

2 Background and Methods

2.1 Research Background

In this study, we selected three semantic similarity based models which had been found to perform strongly in previous research, and which were easy to re-implement: SENSE COLLOCATION (Moldovan et al., 2004), CONSTITUENT SIMILARITY (Kim and Baldwin, 2005) and CO-TRAINING, e.g. using SENSE COLLOCATION or CONSTITUENT SIMILARITY (Kim and Baldwin, 2007). These approaches were evaluated over a 7-way classification using open-domain data from the nominal pair interpretation task of SEMEVAL-2007 (Girju et al., 2007). We test their performance in both 7-way and binary-class classification settings.

2.2 Sense Collocation Method

The SENSE COLLOCATION method of Moldovan et al. (2004) is based on the pair of word senses of NC constituents. The basic idea is that NCs which have the same or similar sense collocation tend to have the same SR. As an example, *car factory* and *automobile factory* share the conventional interpretation of MAKE, which is predicted by *car* and *automobile* having the same sense across the two NCs, and *factory* being used with the same sense in each instance. This intuition is formulated in Equations 1 and 2 below.

The probability $P(r|f_i f_j)$ (simplified to $P(r|f_{ij})$) of a SR r for word senses f_i and f_j is calculated based on simple maximum likelihood estimation:

$$P(r|f_{ij}) = \frac{n(r, f_{ij})}{n(f_{ij})} \quad (1)$$

The preferred SR r^* for the given sense combination is that which maximises the probability:

$$\begin{aligned} r^* &= \text{argmax}_{r \in R} P(r|f_{ij}) \\ &= \text{argmax}_{r \in R} P(f_{ij}|r)P(r) \end{aligned} \quad (2)$$

2.3 Constituent Similarity Method

The intuition behind the CONSTITUENT SIMILARITY method is similar to the SENSE COLLOCATION method, in that NCs made up of similar words tend to share the same SR. The principal difference is that it doesn't presuppose that we know the word sense of each constituent word (i.e. the similarity is calculated at the *word* rather than sense level). The method takes the form of a 1-nearest neighbour classifier, with the best-matching training instance for each test instance predicting its SR. For example, we may find that test instance *chocolate milk* most closely matches *apple juice* and hence predict that the SR is MATERIAL.

This idea is formulated in Equation 3 below. Formally, S_A is the similarity between NCs $(N_{i,1}, N_{i,2})$ and $(B_{j,1}, B_{j,2})$:

$$\begin{aligned} &S_A((N_{i,1}, N_{i,2}), (B_{j,1}, B_{j,2})) = \\ &\frac{((\alpha S1 + S1) \times ((1-\alpha)S2 + S2))}{2} \end{aligned} \quad (3)$$

where $S1$ is the modifier similarity (i.e. $S(N_{i,1}, B_{j1})$) and $S2$ is the head noun similarity (i.e. $S(N_{i,2}, B_{j2})$); $\alpha \in [0, 1]$ is a weighting factor. The similarity scores are calculated across the bag of WordNet senses (without choosing between

them) using the method of Wu and Palmer (1994) as implemented in `WordNet::Similarity` (Patwardhan et al., 2003). This is done for each pairing of WordNet senses of the two words in question, and the overall lexical similarity is calculated as the average across the pairwise sense similarities.

2.4 Co-Training by Sense Collocation

Co-training by sense collocation (SCOLL CO-TRAINING) is based on the SENSE COLLOCATION method and lexical substitution (Kim and Baldwin, 2007). It expands the set of training NCs from a relatively small number of manually-tagged seed instances. That is, it makes use of extra training instances fashioned through a bootstrap process. For example, assuming *automobile factory* with the SR MAKE were a seed instance, NCs generated from synonyms, hypernyms and sister words of its constituents would be added as extra training instances, with the same SR of MAKE. That is, we would add *car factory* (**SYNONYM**), *vehicle factory* (**HYPERNYM**) and *truck factory* (**SISTER WORD**), for example. Note that the substitution takes place only for one constituent at a time to avoid extreme variation.

2.5 Co-training by Constituent Similarity

Co-training by Constituent Similarity (CS CO-TRAINING) is also a co-training method, but based on CONSTITUENT SIMILARITY rather than SENSE COLLOCATION. The basic idea is that when NCs are interpreted using the CONSTITUENT SIMILARITY method, the predictions are more reliable when the lexical similarity is higher. Hence, we progressively reduce the similarity threshold, and incorporate higher-similarity instances into our training data earlier in the bootstrap process. That is, we run the CONSTITUENT SIMILARITY method and acquire NCs with similarity equal to or greater than a fixed threshold. Then in the next iteration, we add the acquired NCs into the training dataset for use in classifying more instances. As a result, in each step, the number of training instances increases monotonically. We "cascade" through a series of decreasing similarity thresholds until we reach a saturation point. As our threshold, we used a starting value of 0.90, which was decremented down to 0.65 in steps of 0.05.

Method	Description
SCOLL	sense collocation
SCOLL_{CT}	sense collocation + SCOLL co-training
CSIM	constituent similarity
CSIM +SCOLL_{CT}	constituent similarity + SCOLL co-training
HYBRID	SCOLL + CSIM + SCOLL_{CT}
CSIM_{CT}	constituent similarity + CSIM co-training

Table 1: Systems used in our experiments

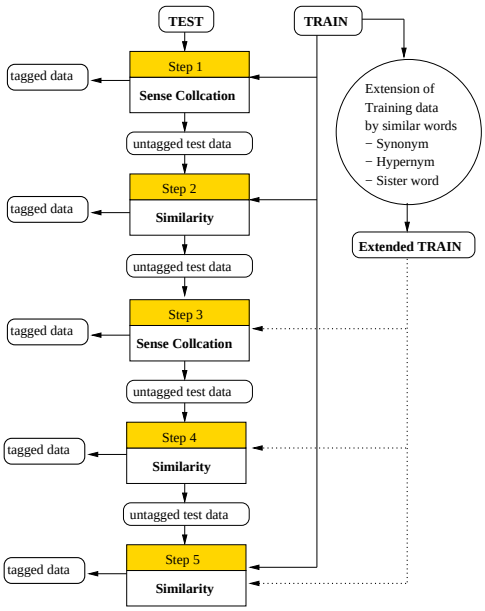


Figure 1: Architecture of the HYBRID method

3 Systems and Architectures

We tested the original methods of Moldovan et al. (2004) and Kim and Baldwin (2005), and combined them with the co-training methods of Kim and Baldwin (2007) to come up with six different hybrid systems for evaluation, as detailed in Table 1. To build the classifiers, we used the TIMBL5.0 memory-based learner (Daelemans et al., 2004).

The HYBRID method consists of five interpretation steps. The first step is to use the SENSE COLLOCATION method over the original training data. When the sense collocation of the test and training instances is the same, we judge the predicted SR to be correct. The second step is to apply the CONSTITUENT SIMILARITY method over the original training data. In order to confirm that the predicted SR is correct, we use a threshold of 0.8 to interpret the test instances. The third step is to apply SENSE COLLOCATION over the expanded train-

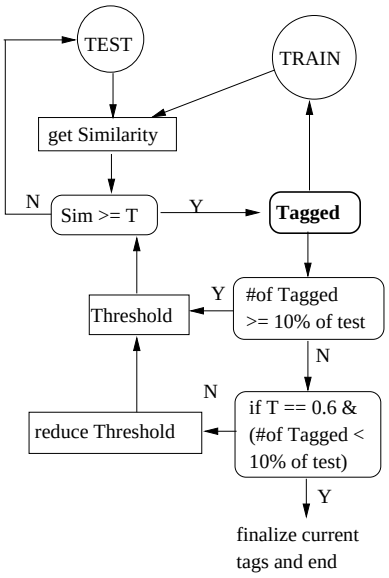


Figure 2: Architecture of the CSIM$_{CT}$ system

ing data through the advent of hypernyms and sister words, using the SCOLL CO-TRAINING method. This step benefits from a larger amount of training data (17,613 vs. 937). The fourth step is to apply the CONSTITUENT SIMILARITY method (EXTCS) over the consolidated training data, with the threshold unchanged at 0.8. The final step is to apply the CONSTITUENT SIMILARITY (CSTT) method over the combined training data without any restriction on the threshold (to guarantee a SR prediction for every test instance). We select SRs from the training instances whose similarity is higher than the original training data and expanded training data. However, since the generated training instances are more likely to contain errors, we apply a linear weight of 0.8 to the similarity values for the expanded training instances. This gives preferential treatment to predictions based on the original training instances. Note that this weight was based on analysis of the error rate in the expanded training instances. In previous work (Kim and Baldwin, 2007), we found the overall classification accuracy rate after the first iteration to be 70-80%. Hence, we settled on a weight of 0.8.

The CSIM$_{CT}$ system is based solely on the CONSTITUENT SIMILARITY method with cascading. We perform iterative CS co-training as described in Section 2.5, with the slight variation that we hold off on reducing the threshold if less than 10% of the test instances are tagged on a given iteration, giving other test instances a chance to be tagged at a higher threshold level relative to newly generated training instances. The residue of test instances on completion of the final iteration (threshold = 0.6) are tagged according to the best-matching training instance, irrespective of the magnitude of the similarity.

	Binary			7-way		
SR	Test	Train	Train*	Test	Train	Train*
CE	80	136	2,588	36	71	1,854
IA	78	135	1,400	36	68	1,001
PP	93	126	2,591	55	78	2,089
OE	81	136	3,085	35	52	1,560
TT	71	129	2,994	27	50	1,718
PW	72	138	2,577	28	64	1,510
CC	74	137	2,378	37	63	1,934
Total	549	937	17,613	254	446	11,664

Table 3: Number of instances associated with each SR (Train* is the number of expanded train instances)

4 Data

We used the dataset from the SEMEVAL-2007 nominal pair interpretation task, which is based on 7 SRs: CAUSE-EFFECT (CE), INSTRUMENT-AGENCY (IA), PRODUCT-PRODUCER (PP), ORIGIN-ENTITY (OE), THEME-TOOL (TT), PART-WHOLE (PW), CONTENT-CONTAINER (CC). The task in SEMEVAL-2007 was to identify the compatibility of a given SR for each test instances using word senses retrieved from WORDNET 3.0 (Fellbaum, 1998) and queries. Table 2 shows the definition of the SRs.

In our research, we interpret the dataset in two ways: (1) as a **binary classification** task for each SR based on the original data; and (2) as a **7-way classification** task, combining together all positive test and training instances for each of the 7 SR datasets into a single dataset. Hence, the size of the dataset for 7-way classification is much smaller than that of the original dataset. We also expand the training instances using SCOLL CO-TRAINING. Table 3 describes the number of test and train instances for NC interpretation for the binary and 7-way classification tasks.

Our analysis shows that only 5 NCs are repeated

Semantic relation	Definition	Examples
Cause-Effect (**CE**)	N_1 is the cause of N_2	*virus flu, hormone growth*
Instrument-Agency (**IA**)	N_1 is the instrument of N_2; N_2 uses N_1	*laser printer, axe murderer*
Product-Producer (**PP**)	N_1 is a product of N_2; N_2 produces N_1	*honey bee, music clock*
Origin-Entity (**OE**)	N_1 is the origin of N_2	*bacon grease, desert storm*
Theme-Tool (**TT**)	N_2 is intended for N_1	*reorganization process, copyright law*
Part-Whole (**PW**)	N_1 is part of N_2	*table leg, daisy flower*
Content-Container (**CC**)	N_1 is stored or carried inside N_2	*apple basket, wine bottle*

Table 2: The set of 7 semantic relations, where N_1 is the modifier and N_2 is the head noun

across multiple SR datasets (i.e. occur as an instance in more than one of the 7 datasets), none of which occur as positive instances for multiple SRs. As such, no NC instances in the 7-way classification task end up with a multiclass classification. Also note that some of NCs are contained within ternary or higher-order NCs: 40 test NCs and 81 training NCs for the binary classification task, and 24 test NCs and 42 training NCs for the 7-way classification task. For these NCs, we extracted a "base" binary NC based on the provided bracketing. The following are examples of extraction of binary NCs from ternary or higher-order NCs.

((billiard table) room) → table room
(body (bath towel)) → body towel

In order to extract a binary NC, we take the head noun of each embedded NC and combine this with the corresponding head noun or modifier. E.g., *table* is the head noun of *billiard table*, which combines with the head noun of the complex NC *room* to form *table room*.

5 Experiment 1: 7-way classification

Our first experiment was carried out over the 7-way classification task—i.e. all 7 SRs in a single classification task—using the 6 systems from Section 3. In our results in Table 4, we use the system categories from SEMEVAL-2007 of **A4** and **B4**, where A4 systems use none of the provided word senses, and B4 systems use the word senses.[3] We categorized our systems into these two groups in order to evaluate them separately within the bounds of the original SEMEVAL-2007 task. In each case, the baseline is a majority class classifier.

[3]In the original SEMEVAL-2007 task, there were two further categories, which incorporated the "query" with or without the sense information.

Class	Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{A}$
–	Majority				.217
A4	CSIM	.518	.522	**.449**	**.528**
	CSIM_{CT}	.517	.511	.426	.522
B4	SCOLL	.705	.444	.477	.496
	SCOLL_{CT}	.646	.466	**.498**	.508
	CSIM +SCOLL_{CT}	.523	.520	.454	**.528**
	HYBRID	.500	.505	.416	.516

Table 4: Experiment 1: Results ($\mathcal{P}$=precision, $\mathcal{R}$=recall, $\mathcal{F}_1$=F-score, $\mathcal{A}$=accuracy)

Step	Method	Tagged	$\mathcal{A}_i$	Untagged
1	SCOLL	12	1.000	242
2	CSIM	57	.719	185
3	extSCOLL	0	.000	185
4	extCSIM	78	.462	107
5	$\text{CSIM}_{\text{REST}}$	107	.393	0

Table 5: Experiment 1: Classifications for each step of the HYBRID method (CS_{REST}=the final application of CS over the remaining test instances, $\mathcal{A}_i$=accuracy for classifications made at step i)

Tables 5 and 6 show the results at each step for the HYBRID and CSIM_{CT} methods, respectively. As each method proceeds, the amount of tagged data increases but the classification accuracy of the system decreases, due to the inclusion of increasingly noisy training instances in the previous step. The performance of each individual relation is shown in Figure 3, which largely mirrors the findings of the systems in the original SEMEVAL-2007 task in terms of the relative difficulty to predict each of the 7 SRs.

6 Experiment 2: binary classification

In the second experiment, we performed a separate binary classification task for each of the 7 SRs, in the manner of the original SEMEVAL-2007 task. Table 7 shows the three baselines provided by the SEMEVAL-2007 organisers and performance of our

Iteration	θ	Tagged	$\mathcal{A}_i$	Untagged
1	.90	29	.897	225
2	.85	12	.750	213
3	.80	31	.613	182
4	.75	43	.535	139
5	.70	63	.540	76
6	.65	26	.346	50
7	<.65	49	.250	1

Table 6: Experiment 1: Classifications at each step of the CSIM$_{\mathrm{CT}}$ method (θ=threshold, $\mathcal{A}_i$=accuracy for classifications made at iteration i)

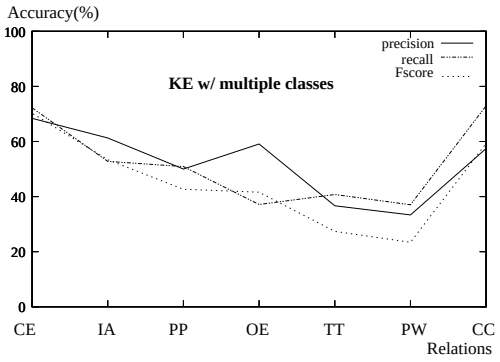


Figure 3: Experiment 1: Performance over each SR (CSIM +SCOLL$_{\mathrm{CT}}$ method)

6 systems. We also present the best-performing system within each group from the SEMEVAL-2007 task. The methods for computing the baselines are described in Girju et al. (2007).

As with the first experiment, we analyzed the number of tagged instances and accuracy for the HYBRID and CSIM$_{\mathrm{CT}}$ methods, as shown in Tables 8 and 9, respectively. The overall results are similar to those for the 7-way classification task.

Figures 4 and 5 show the performance for positive and negative classifications for each individual SR. The performance when the classifier outputs are mapped onto the 7-way classification task are similar to those in Figure 3.

7 Discussion and Conclusion

We compared the performance of the 6 systems in Tables 4 and 7 over the 7-way and binary classification tasks, respectively. The performance of all methods exceeded the baseline. The CONSTITUENT SIMILARITY (CSIM) system performed the best in group A4 and CONSTITUENT SIMILARITY + SCOLL$_{\mathrm{CT}}$ (CSIM +SCOLL$_{\mathrm{CT}}$) system performed the best in group B4 for both classification tasks. In general, the performance of CONSTITUENT SIMILARITY is marginally better than that of SENSE COLLOCATION. Also, the utility of co-training is confirmed by it outperforming both CONSTITUENT SIMILARITY and SENSE COLLOCATION.

Class	Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{A}$
–	All True	.485	1.000	.648	.485
–	Probability	.485	.485	.485	.517
–	Majority	.813	.429	.308	.570
A4	Best	.661	.667	.648	.660
	CSIM	.632	.628	**.627**	**.650**
	CSIM$_{\mathrm{CT}}$	.615	.557	.578	.627
B4	Best	.797	.698	.724	.763
	SCOLL	.672	.584	.545	.634
	SCOLL$_{\mathrm{CT}}$	.602	.571	.554	.619
	CSIM +SCOLL$_{\mathrm{CT}}$	.660	.657	**.654**	**.669**
	HYBRID	.617	.568	.587	.625

Table 7: Experiment 2: Binary classification results ($\mathcal{P}$=precision, $\mathcal{R}$=recall, $\mathcal{F}_1$=F-score, $\mathcal{A}$=accuracy)

Step	Method	Tagged	$\mathcal{A}_i$	Untagged
1	SCOLL	21	.810	526
2	CSIM	106	.689	420
3	extSCOLL	0	.000	420
4	extCSIM	61	.607	359
5	CSIM$_{\mathrm{REST}}$	359	.619	0

Table 8: Experiment 2: Classifications for each step of the HYBRID method (CS$_{\mathrm{REST}}$=the final application of CS over the remaining test instances, $\mathcal{A}_i$=accuracy for classifications made at step i)

In order to compare the original methods with the hybrid methods, we observed that the original methods, SCOLL and K, and their co-training variants performed consistently better than the hybrid methods, HYBRID and CSIM$_{\mathrm{CT}}$. We found that the combination of the methods lowers overall performance. We also found that the number of training instances contributes to improved performance, predictably in the sense that the methods are supervised, but encouraging in the sense that the extra training data is generated automatically. As expected, the step-wise performance of HYBRID and CSIM$_{\mathrm{CT}}$ degrades with each iteration, although there were instances where the performance didn't drop from one iteration to the next (e.g. iteration 3 = 59.46% vs. iteration 4 = 72.23% in Experiment 2). This confirms

Iteration	θ	Tagged	$\mathcal{A}_i$	Untagged
1	.90	21	.810	526
2	.85	52	.726	474
3	.80	56	.714	418
4	.75	74	.595	344
5	.70	101	.722	243
6	.65	222	.572	21
7	$<$.65	21	.996	0

Table 9: Experiment 2: Classifications at each step of the CSIM$_{\text{CT}}$ method (θ=threshold, $\mathcal{A}_i$=accuracy for classifications made at iteration i)

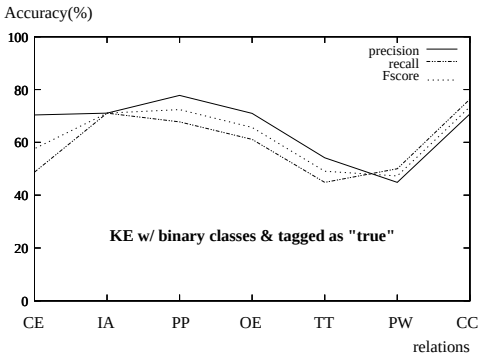


Figure 4: TPR for each SR for the binary task (positive instances, CSIM +SCOLL$_{\text{CT}}$ method)

our expectation that: (a) the similarity threshold is strongly correlated with the quality of the resultant data; and (b) the method is susceptible to noisy training data.

Our performance comparison over the binary classification task from the SEMEVAL-2007 task shows that our 6 systems performed below the best performing system in the competition, to varying degrees. This is partly because the methods were originally designed for multi-way (positive) classification and require adjustment for the binary task reformulation, although their performance is competitive.

Finally, comparing the SCOLL and CSIM methods, we found that the methods interpret SRs with 100% accuracy when the sense collocations are found in both the test and training data. However, the CSIM method is more sensitive than the SCOLL method to variation in the sense collocations, which leads to better performance. Also, the CSIM method interprets NCs with high accuracy when the computed similarity is sufficiently high (e.g. with similarity ≥ 0.9 the accuracy is 89.7%). Another benefit

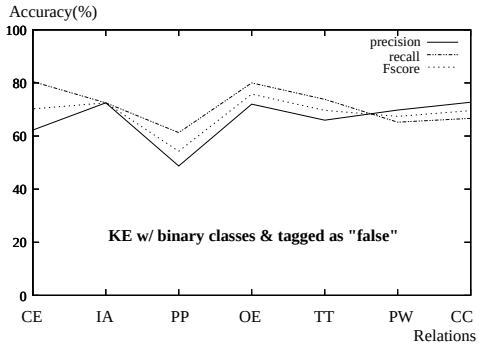


Figure 5: TNR for each SR for the binary task (negative instances, CSIM +SCOLL$_{\text{CT}}$ method)

of this method is that it interprets NCs without word sense information. As a result, we conclude that the CSIM method is more flexible and robust. One possible weakness of CSIM is its reliance on the similarity measure.

8 Conclusions and Future Work

In this paper, we have benchmarked and hybridised existing NC interpretation methods over data from the SEMEVAL-2007 nominal pair interpretation task. In this, we have established guidelines for the use of the different methods, and also for the reinterpretation of the SEMEVAL-2007 data as a more conventional multi-way classification task. We confirmed that CONSTITUENT SIMILARITY is the best method due to its insensitivity to varied sense collocations. We also confirmed that co-training improves the performance of the methods by expanding the number of training instances.

Looking to the future, there is room for improvement for all the methods through such factors as threshold tweaking and expanding the training instances further.

References

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pp. 805–810, Acapulco, Mexico.

Ken Barker and Stan Szpakowicz. 1998. Semi-automatic recognition of noun modifier relationships. In *Proceedings of the 17th International Conference*

on Computational Linguistics, pp. 96–102, Montreal, Canada.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. *TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide*. ILK Technical Report 04-02.

Pamela Downing. 1977. On the Creation and Use of English Compound Nouns. *Language*, 53(4):810–842.

James Fan and Ken Barker and Bruce W. Porter. 2003. The knowledge required to interpret noun compounds. In *In Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 1483–1485.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

Timothy W. Finin. 1980. *The Semantic Interpretation of Compound Nominals*. Ph.D. thesis, University of Illinois.

Roxana Girju. 2007. Improving the Interpretation of Noun Phrases with Cross-linguistic Information. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 568–575, Prague, Czech Republic.

Roxana Girju and Preslav Nakov and Vivi Nastase and Stan Szpakowicz and Peter Turney and Deniz Yuret. 2007. SemEval-2007 Task 04: Classification of Semantic Relations between Nominals. In *Proceedings of the 4th Semantic Evaluation Workshop (SemEval-2007)*, Prague, Czech Republic, pp.13–18.

Su Nam Kim and Timothy Baldwin. 2005. Automatic interpretation of Noun Compounds using WordNet similarity. In *Proceedings of the 2nd International Joint Conference On Natural Language Processing*, pp. 945–956, JeJu, Korea.

Su Nam Kim and Timothy Baldwin. 2006. Interpreting Semantic Relations in Noun Compounds via Verb Semantics. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics (COLING/ACL-2006)*. pp. 491–498, Sydney, Australia.

Su Nam Kim and Timothy Baldwin. 2007. Interpreting Noun Compound Using Bootstrapping and Sense Collocation. In *Proceedings of the Pacific Association for Computational Linguistics (PACLING)*, pp. 129–136, Melbourne, Australia.

Maria Lapata. 2002. The disambiguation of nominalizations. *Computational Linguistics*, 28(3):357–388.

Judith Levi. 1979. The syntax and semantics of complex nominals. In *The Syntax and Semantics of Complex Nominals*. New York:Academic Press.

Dan Moldovan, Adriana Badulescu, Marta Tatu, Daniel Antohe, and Roxana Girju. 2004. Models for the semantic classification of noun phrases. In *Proceedings of the HLT-NAACL 2004 Workshop on Computational Lexical Semantics*, pp. 60–67, Boston, USA.

Preslav Nakov and Marti Hearst. 2006. Using Verbs to Characterize Noun-Noun Relations. In *Proceedings of the 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA)*, Bularia.

Diarmuid Ó Séaghdha and Ann Copestake. 2007. Co-occurrence Contexts for Noun Compound Interpretation. In *Proc. of the ACL-2007 Workshop on A Broader Perspective on Multiword Expressions*, Prague, Czech Republic.

Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*.

Barbara Rosario and Hearst Marti. 2001. Classifying the Semantic Relations in Noun Compounds via a Domain-Specific Lexical Hierarchy. In *In Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP-2001)*, 82–90.

Karen Sparck Jones. 1983. Compound noun interpretation problems. Computer Speech Processing, Frank Fallside and William A. Woods, Prentice-Hall, Englewood Cliffs, NJ.

Lucy Vanderwende. 1994. Algorithm for automatic interpretation of noun sequences. In *Proceedings of the 15th Conference on Computational linguistics*, pp. 782–788.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 133–138, Las Cruces, USA.