

Hitoshi Isahara
Kyoko Kanzaki (Eds.)

LNAI 7614

Advances in Natural Language Processing

8th International Conference on NLP, JapTAL 2012
Kanazawa, Japan, October 2012
Proceedings

JapTAL  2012

 Springer

Lecture Notes in Artificial Intelligence 7614

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

Hitoshi Isahara Kyoko Kanzaki (Eds.)

Advances in Natural Language Processing

8th International Conference on NLP, JapTAL 2012
Kanazawa, Japan, October 22-24, 2012
Proceedings



Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Hitoshi Isahara
Toyohashi University of Technology
Information and Media Center
1-1 Hibarigaoka, Tenpakucho
Toyohashi 441-8580, Japan
E-mail: isahara@tut.jp

Kyoko Kanzaki
Toyohashi University of Technology
Information and Media Center
1-1 Hibarigaoka, Tenpakucho
Toyohashi 441-8580, Japan
E-mail: kyoko.kanzaki@gmail.com

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-33982-0 e-ISBN 978-3-642-33983-7
DOI 10.1007/978-3-642-33983-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012948107

CR Subject Classification (1998): I.2, H.3, H.4, H.5, H.2, J.1, I.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

© Springer-Verlag Berlin Heidelberg 2012

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

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

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

Printed on acid-free paper

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

Preface

The 8th International Conference on Natural Language Processing (JapTAL 2012) took place in the city of Kanazawa, a leading tourist destination where 7 million tourists visit every year. It was a great castle town ruled by an influential leader from the seventeenth century to the second half of the nineteenth century. Kanazawa has not suffered from any war devastation or big natural disaster up to now. Therefore, it maintains rows of historical houses and inherits traditional handicrafts and traditional performing arts.

JapTAL was the eighth in the series of the TAL conferences, following IceTAL 2010 (Reykjavik, Iceland), GoTAL 2008 (Gothenburg, Sweden), FinTAL 2006 (Turku, Finland), EsTAL 2004 (Alicante, Spain), PorTAL 2002 (Faro, Portugal), VexTAL 1999 (Venice, Italy), and FracTAL 1997 (Besancon, France). This was the first time that the TAL conference was held outside Europe.

The main purpose of the TAL conference series is to bring together scientists representing linguistics, computer science, and related fields, sharing a common interest in the advancement of computational linguistics and natural language processing.

The conference consists of invited talks, oral and poster presentations, and special sessions on the applications and theory of natural language processing and related areas. It provides excellent opportunities for the presentation of interesting new research results and discussions about them, leading to knowledge transfer and the generation of new ideas.

In the reviewing process of the main conference track, we received 42 submissions. Among them, we selected 27 submissions as long papers and five submissions as short papers. Therefore, the acceptance ratio for the long papers is 64% and the total acceptance ratio including short papers is 76%.

We had two special sessions, i.e., “Game and NLP” and “Student/Young Researcher Session.” Papers submitted for these sessions were treated separately and accepted papers were presented in each session but they are not included in this volume.

As conference organizers of JapTAL 2012, we would like to thank all PC members who reviewed submissions very tight schedule and all local staff who helped us during the preparation of JapTAL. We would like to thank the International Exchange Program of the National Institute of Information and Communications Technology (NICT) for its support of JapTAL 2012.

Hitoshi Isahara
Kyoko Kanzaki

Organization

Program Chairs

Hitoshi Isahara
Kyoko Kanzaki

Program Committee

Takako Aikawa	Microsoft Research, USA
Johan Bos	University of Groningen, The Netherlands
Pierrette Bouillon	Geneva University, Switzerland
Caroline Brun	Xerox Corporation, France
Sylviane Cardey	University of Franche-Comté, France
Key-Sun Choi	KAIST, Korea
Christiane Fellbaum	Princeton University, USA
Filip Ginter	University of Turku, Finland
Peter Greenfield	University of Franche-Comté, France
Philippe de Groote	INRIA, France
Yurie Iribe	Toyohashi University of Technology, Japan
Katsunori Kotani	Kansai Gaidai University, Japan
Krister Lindén	University of Helsinki, Finland
Hrafn Loftsson	Reykjavik University, Iceland
Qing Ma	Ryukoku University, Japan
Bente Maegaard	University of Copenhagen, Denmark
Mathieu Morey	Nanyang Technological University, Singapore
Masayuki Okabe	Toyohashi University of Technology, Japan
Guy Perrier	INRIA, France
Kiyooki Shirai	JAIST, Japan
Virach Sornlertlamvanich	NECTEC, Thailand
Koichi Takeuchi	Okayama University, Japan
Midori Tatsumi	Toyohashi University of Technology, Japan
Izabella Thomas	University of Franche-Comté, France
Noriko Tomuro	DePaul University, USA
Masatoshi Tsuchiya	Toyohashi University of Technology, Japan
Jose Luis Vicedo	University of Alicante, Spain
Simo Vihjanen	Lingsoft Ltd., Finland
Xiaohong Wu	Qinghai University for Nationalities, China
Kazuhide Yamamoto	Nagaoka University of Technology, Japan
Yujie Zhang	Beijing Jiaotong University, China
Tiejun Zhao	Harbin Institute of Technology, China

Table of Contents

Machine Translation

The Impact of Crowdsourcing Post-editing with the Collaborative Translation Framework	1
<i>Takako Aikawa, Kentaro Yamamoto, and Hitoshi Isahara</i>	
Translation of Quantifiers in Japanese-Chinese Machine Translation	11
<i>Shaoyu Chen and Tadahiro Matsumoto</i>	
Toward Practical Use of Machine Translation	23
<i>Hitoshi Isahara</i>	
Phrase-Level Pattern-Based Machine Translation Based on Analogical Mapping Method	28
<i>Jun Sakata, Masato Tokuhisa, and Jin'ichi Murakami</i>	

Multilingual Issues

Parallel Texts Extraction from Multimodal Comparable Corpora	40
<i>Haithem Afli, Loïc Barrault, and Holger Schwenk</i>	
A Reliable Communication System to Maximize the Communication Quality	52
<i>Gan Jin and Natallia Khatseyeva</i>	
DANIEL: Language Independent Character-Based News Surveillance	64
<i>Gaël Lejeune, Romain Brixtel, Antoine Doucet, and Nadine Lucas</i>	
OOV Term Translation, Context Information and Definition Extraction Based on OOV Term Type Prediction	76
<i>Jian Qu, Akira Shimazu, and Le Minh Nguyen</i>	
Exploiting a Web-Based Encyclopedia as a Knowledge Base for the Extraction of Multilingual Terminology	88
<i>Fatima Sadat</i>	
Segmenting Long Sentence Pairs to Improve Word Alignment in English-Hindi Parallel Corpora	97
<i>Jyoti Srivastava and Sudip Sanyal</i>	
Shallow Syntactic Preprocessing for Statistical Machine Translation	108
<i>Hoai-Thu Vuong, Dao Ngoc Tu, Minh Le Nguyen, and Vinh Van Nguyen</i>	

Resources

Linguistic Rules Based Approach for Automatic Restoration of Accents on French Texts	118
<i>Paul Brillant Feuto Njonko, Sylviane Cardey-Greenfield, and Peter Greenfield</i>	
Word Clustering for Persian Statistical Parsing	126
<i>Masood Ghayoomi</i>	
Building a Lexically and Semantically-Rich Resource for Paraphrase Processing	138
<i>Wannachai Kampeera and Sylviane Cardey-Greenfield</i>	
Tagset Conversion with Decision Trees	144
<i>Bartosz Zaborowski and Adam Przepiórkowski</i>	
Fitting a Round Peg in a Square Hole: Japanese Resource Grammar in GF	156
<i>Elizaveta Zimina</i>	

Semantic Analysis

Arabic Language Analyzer with Lemma Extraction and Rich Tagset	168
<i>Ahmed H. Aliwy</i>	
Tracking Researcher Mobility on the Web Using Snippet Semantic Analysis	180
<i>Jorge J. García Flores, Pierre Zweigenbaum, Zhao Yue, and William Turner</i>	
Semantic Role Labelling without Deep Syntactic Parsing	192
<i>Konrad Gołuchowski and Adam Przepiórkowski</i>	
Temporal Information Extraction with Cross-Language Projected Data	198
<i>Przemysław Jarzębowski and Adam Przepiórkowski</i>	
Word Sense Disambiguation Based on Example Sentences in Dictionary and Automatically Acquired from Parallel Corpus	210
<i>Pulkit Kathuria and Kiyooki Shirai</i>	
A Study on Hierarchical Table of Indexes for Multi-documents	222
<i>Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu</i>	
Finding Good Initial Cluster Center by Using Maximum Average Distance	228
<i>Samuel Sangkon Lee and Chia Y. Han</i>	

Applying a Burst Model to Detect Bursty Topics in a Topic Model	239
<i>Yusuke Takahashi, Takehito Utsuro, Masaharu Yoshioka, Noriko Kando, Tomohiro Fukuhara, Hiroshi Nakagawa, and Yoji Kiyota</i>	

UDRST: A Novel System for Unlabeled Discourse Parsing in the RST Framework	250
<i>Ngo Xuan Bach, Nguyen Le Minh, and Akira Shimazu</i>	

Sentiment Analysis

Long-Term Goal Discovery in the Twitter Posts through the Word-Pair LDA Model	262
<i>Dandan Zhu, Yusuke Fukazawa, Eleftherios Karapetsas, and Jun Ota</i>	

Finding Social Relationships by Extracting Polite Language in Micro-blog Exchanges	268
<i>Norinobu Hatamoto, Yoshiaki Kurosawa, Shogo Hamada, Kazuya Mera, and Toshiyuki Takezawa</i>	

Twitter Sentiment Analysis Based on Writing Style	278
<i>Hiroshi Maeda, Kazutaka Shimada, and Tsutomu Endo</i>	

Extraction of User Opinions by Adjective-Context Co-clustering for Game Review Texts	289
<i>Kevin Raison, Noriko Tomuro, Steve Lytinen, and Jose P. Zagal</i>	

Speech and Generation

Automatic Phone Alignment: A Comparison between Speaker- Independent Models and Models Trained on the Corpus to Align	300
<i>Sandrine Brognaux, Sophie Roekhaut, Thomas Drugman, and Richard Beaufort</i>	

A Story Generation System Based on Propp Theory: As a Mechanism in an Integrated Narrative Generation System	312
<i>Shohei Imabuchi and Takashi Ogata</i>	

Automatic Utterance Generation by Keeping Track of the Conversation's Focus within the Utterance Window	322
<i>Yusuke Nishio and Dongli Han</i>	

Author Index	333
-------------------------------	------------

The Impact of Crowdsourcing Post-editing with the Collaborative Translation Framework

Takako Aikawa¹, Kentaro Yamamoto², and Hitoshi Isahara²

¹ Microsoft Research, Machine Translation Team
takakoa@microsoft.com

² Toyohashi University of Technology
yamamoto@lang.cs.tut.ac.jp, isahara@tut.jp

Abstract. This paper presents a preliminary report on the impact of crowdsourcing post-editing through the so-called “Collaborative Translation Framework” (CTF) developed by the Machine Translation team at Microsoft Research. We first provide a high-level overview of CTF and explain the basic functionalities available from CTF. Next, we provide the motivation and design of our crowdsourcing post-editing project using CTF. Last, we present the results from the project and our observations. Crowdsourcing translation is an increasingly popular-trend in the MT community, and we hope that our paper can shed new light on the research into crowdsourcing translation.

Keywords: Crowdsourcing post-editing, Collaborative Translation Framework.

1 Introduction

The output of machine translation (MT) can be used either as-is (i.e., raw-MT) or for post-editing (i.e., MT for post-editing). Although the advancement of MT technology is making raw-MT use more pervasive, reservations about raw-MT still persist; especially among users who need to worry about the accuracy of the translated contents (e.g., government organizations, education institutes, NPO/NGO, enterprises, etc.). Professional human translation from scratch, however, is just too expensive. To reduce the cost of translation while achieving high translation quality, many places use MT for post-editing; that is, use MT output as an initial draft of translation and let human translators post-edit it. Many researchers (both from academia and industry) have been investigating how to optimize the post-editing process and developing tools that can achieve high productivity gains via MT for post-editing.¹

Recently, another type of approach to reduce the cost of translation has surfaced; namely, crowdsourcing translation. Crowdsourcing translation started as a method to create training/evaluation data for statistical machine translation (SMT). For instance, with Amazon’s Mechanical Turk, one can create a huge amount of bilingual corpus

¹ See Allen (2003, 2005)[1][2], O’Brien (2005)[3], Guerberof (2009a/b)[4][5], Koehn and Haddow (2009)[6], for instance.

data to build a new SMT system in a relatively inexpensive and quick way (Ambati et al. (2010)[7], Zaidan and Callison-Burch (2011)[8], Ambati and Vogel (2011)[9]).² This paper introduces a new way of crowdsourcing translation. Our approach is unique in that it focuses on post-editing and uses a different platform; namely, the Collaborative Translation Framework (CTF) developed by the Machine Translation team at Microsoft Research. For our project, we used foreign students at Toyohashi University of Technology as editors and asked them to post-edit the MT output of the university's English websites (<http://www.tut.ac.jp/english/introduction/>) via Microsoft Translator (<http://www.microsofttranslator.com>) into their own languages using the CTF functionalities. This paper is a preliminary report on the results from this project. The organization of the paper is as follows: Section 2 provides a high level overview of CTF while describing various functionalities associated with CTF. Section 3 presents the design of our crowdsourcing project using Toyohashi University of Technology websites. Section 4 presents a preliminary report on the results from the project and Section 5 provides our concluding remarks.

2 Collaborative Translation Framework (CTF)

As mentioned at the outset of the paper, CTF has been developed by the Machine Translation team at Microsoft Research. CTF aims to create an environment where MT and humans can help each other to improve translation quality in an effective way. One of the prominent functionalities of CTF is to allow users to modify or edit the MT output from Microsoft Translator. Thus, with CTF, we can utilize the power of crowdsourcing to post-edit MT output. There are other types of functionalities associated with CTF, and in the following subsections, we describe these in more detail.

2.1 Basic Functionalities of CTF

CTF functionalities have been fully integrated into Microsoft Translator's Widget (<http://www.microsofttranslator.com/widget>), and one can experience how CTF works by visiting any website(s) with this Widget.³ For instance, let us look at Figure 1, which is the snapshot of the Widget on the English homepage at Toyohashi University of Technology (<http://www.tut.ac.jp/english/introduction/>). With this Widget, users (or visitors of this website) can translate the entire web site automatically into their own languages; select their target languages (in Figure 1, Japanese is being

² Amazon's Mechanical Turk has been also used for creating different types of data as well. For instance, see Callison-Burch (2009) [10] and Higgins et.al (2010)[11].

³ CTF functionalities can also be called via Microsoft Translator's public API's. For more details on Microsoft Translator's API's, visit <http://msdn.microsoft.com/en-us/library/dd576287>

selected) and click the translate button (red-circled), so that the entire page can be translated into that selected target language instantly as shown in Figure 2.⁴

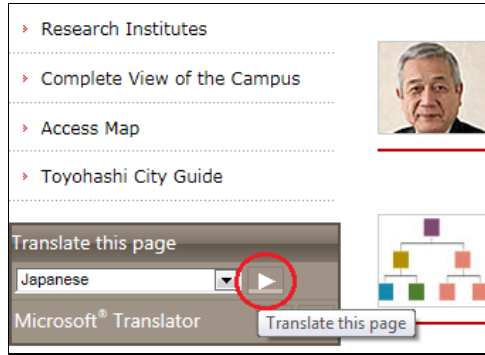



Fig. 1. Widget on the Toyohashi University of Technology website (English)



Fig. 2. Translated into Japanese

Besides the webpage translation described above, CTF functionalities integrated into the Widget can offer users various types of controls. First, by hovering over an individual sentence, users can evoke the CTF user interface (UI) (see Figure 3), and inside the CTF UI, users can see the original sentence and the editorial window where the MT output can be modified as shown in Figure 3.

Second, CTF allows users to see edits from other users. For instance, in Figure 4, the first string next to the icon  is the MT output and the two translations below are the alternative translations provided by other users.

⁴ Microsoft Translator currently supports 38 languages.

See <http://www.microsofttranslator.com/help/> for the list of languages supported by Microsoft Translator.



Fig. 3. The Edit control inside the CTF UI

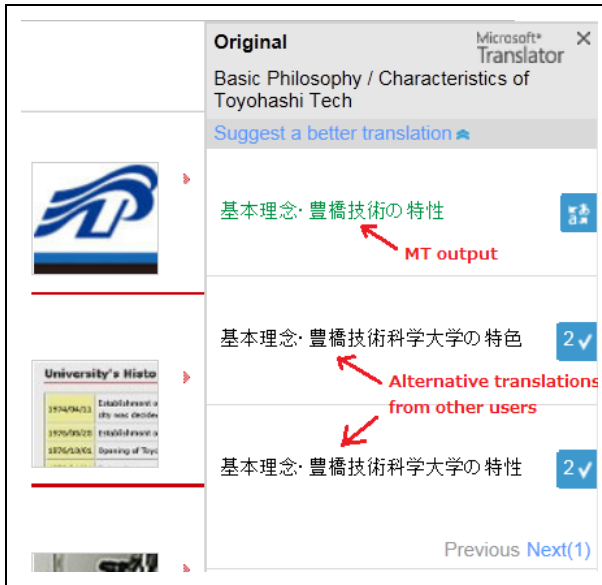


Fig. 4. Alternative translations from other users

Third, users can modify the alternative translations from other users (as well as MT output) as shown in Figure 5.

Note that users can report spam or bad translations by clicking the [Report] button. Furthermore, they can vote for alternative translations if they wish. For instance, the numbers next to the alternative translations in Figure 4/Figure5 above (i.e., “2”) indicate that these translations have already received 2 votes from users.

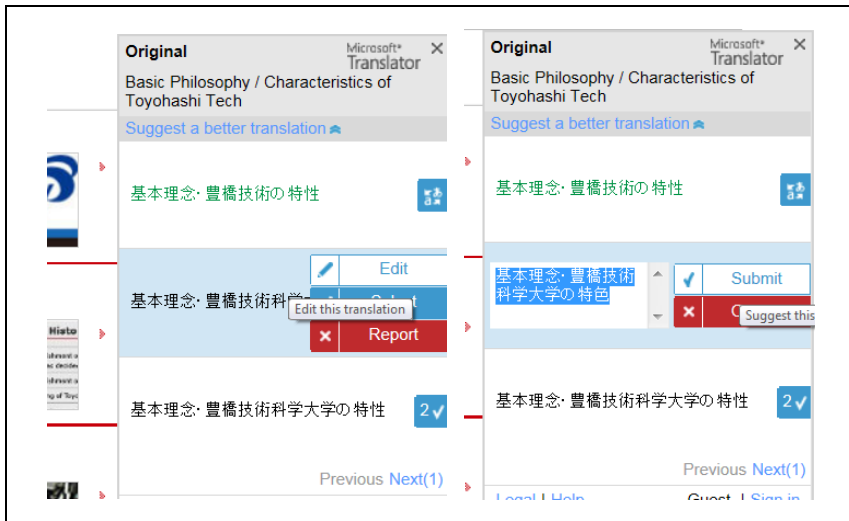


Fig. 5. Post-edit alternative translations

Last but not least, CTF allows a web owner to delegate a set of people as moderators and/or translators. This functionality is critical to our crowdsourcing project and is described in detail in 2.2 below.

2.2 Role-Assignment

One of the biggest concerns for crowdsourcing translation is the quality assurance of crowd-sourced translation. That is, how can we verify the quality of the edits or the translations coming from anonymous users? To address this concern, CTF allows a web owner to assign “trusted” human translators the role of moderator and/or that of translator.

The moderator role can be assigned to someone who can oversee and moderate the quality of the translations coming from translators, and the translator role can be assigned to individuals who can provide their translations. The edits/translations done by these “trusted” users can overwrite MT output or the edits from anonymous users.⁵ This way, the web owner can have more control over the quality assurance of the crowd-sourced post-edits or translations, and she or he can do the assignment of these moderators and translations easily on the CTF dashboard, which is illustrated in Figure 6.

⁵ The hierarchical order of CTF users is: web owner -> moderator -> translator -> anonymous users. The translations from the web owner, the moderator, or the translator can overwrite MT output but those from anonymous users cannot.

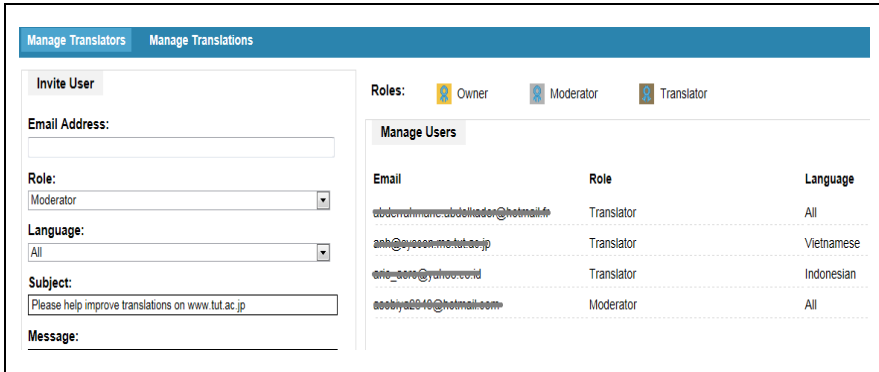


Fig. 6. Role Assignment

3 Crowdsourcing Post-editing: A Case Study at Toyohashi University of Technology

3.1 Background

Toyohashi University of Technology (TUT) has more than 200 foreign students from various countries, and the demand to localize the information on their websites into various languages has always been strong. Yet, localizing the websites using professional translators is just too expensive to do. To make the university information more accessible to these foreign students or to upcoming new students, they created English translations of their websites. However, still lots of foreign students had problems in understanding TUT’s website information because of their language barrier. To overcome this issue, the university decided to conduct this crowdsourcing post-editing project using Microsoft Translator’s CTF.⁶

3.2 Design

For crowdsourcing post-editors, we hired foreign students with different language background at TUT and assigned each of them the role of translator. These students are familiar with the contents of TUT’s websites, which we thought would be a great advantage in post-editing the MT output of TUT’s websites. The total number of student participants was 22, and their language background is provided in Table 1.⁷

⁶ This project is a collaboration between Toyohashi University of Technology and the Machine Translation team at Microsoft Research. See Yamamoto et al. (2011) for our initial report [12].

⁷ Strictly speaking, the total number of student participants was 21 as one of the students edited both Arabic and French MT output.

Table 1. The language background of the student participants

Language	Number of the participants
Arabic	2
Indonesian	2
Portuguese	1
Spanish	4
Chinese (simplified)	6
Vietnamese	2
French	2
German	1
Korean	2

Prior to starting the project, we gave these students a brief introduction on how to use the CTF UI and explained the background of the project. We also provided some specific instructions on how to post-edit MT output. The instructions we provided are summarized below:

- Avoid over-editing: don't try to over-edit if the existing translation(s) (whether they are MT output or other human edits) are grammatical and readable.
- Ignore stylistic differences: don't try to modify stylistic differences unless they are critical or matter for readability.
- Start from scratch: if the quality of MT output is too low, provide your translation from scratch (as opposed to modifying MT output).

**Fig. 7.** Editing alternative translations from other users

It is important to note here that we did not prevent the students from modifying already existing translations provided by other students. For instance, in Figure 7, there are two already existing alternative translations in addition to the MT output. The students are allowed to modify not only the MT output but also any one of these alternative translations if they think it is necessary to modify.

Time-wise, we assigned each student 30 hours of post-editing work on TUT's websites. We conducted this project in November-December, 2011.

4 Results

We have gathered a decent amount of edits from the students as shown in Table 2.

Table 2. Results

(A) Language	(B) Number of sentences edited	(C) Number of edits	(D) Ratio	(E) Average number of edits
Arabic	397	723	45%	361.5
Indonesian	1285	1559	18%	779.5
Portuguese	204	308	34%	308
Spanish	1841	3643	50%	910.75
Chinese (simplified)	1637	2269	28%	378.1
Vietnamese	1341	1929	31%	964.5
French	512	647	21%	323.5
German	147	192	24%	192
Korean	598	707	16%	353.5

Column B refers to the number of original sentences that have been modified. Column C, on the other hand, refers to the total number of edits we got from the student translators. As just mentioned, we did not prevent the students from modifying the edits from others. So some sentences ended up having multiple (alternative) translations, resulting in the gap in number between Column B and Column C. Column D indicates the overall percentage of sentences that have more than one edit(s). For instance, for the case of Arabic, 45% of the original sentences that have more than one edit. Column E indicates how many edits are provided on average by one student (i.e., the numbers of edits divided by the number of students in Table 2).

A couple of observations can be made here. First, the average number of edits varies quite radically depending on the target language. If we simply assume that this average number is pseudo-equivalent to the productivity of post-editing, Vietnamese and Spanish students are expected to be the most productive post-editors, and the German editor the least productive one.⁸ Another interesting phenomenon observed

⁸ It is true that the quality of the MT system also varies depending on the target language, and we can't or shouldn't come up with any conclusion based solely on these numbers. But, let's assume that we can.

here is the high ratio of multiple edits for Spanish. The English->Spanish system is allegedly the best system in terms of the quality of Microsoft Translator, yet this language pair has the highest ratio of multiple post-edits. It is unclear why this is the case, and we would like to investigate this as a future research topic.

5 Concluding Remarks

In this paper, we provided a preliminary report on the results from our crowdsourcing post-editing project using CTF. Using the crowdsourcing power of the foreign students at TUT, we could localize the majority of TUT's English websites into 9 languages within 2 months with inexpensive cost, and we are very happy about this outcome. We also asked these foreign students to give us their verbatim feedback, some of which are provided in Table 3 below. As seen there, the overall feedback from the students is very positive and it is great to see that the students felt the "sense of community" by participating in this project.

Table 3. Verbatim feedback from the participants

Indonesian student	Working as Microsoft translator give me great benefit especially knowing in detail about the content of the TUT website which I previously didn't know.
Spanish studentI have to say that I am very glad that I could be a part of it, and about the project itself, I think it's a great way to Internationalize and attract more overseas studentsSince Machine translation in websites is probably the easiest way of helping readers from different countries communicate, it is often used as first choice, but most of the time it translates incorrectly or some sentences do not make any sense, that's why I believe this was a great opportunity to help improve the webpage by the use of humans.
Spanish student	While doing this job, I was able to realize the complexity of translating without changing the original meaning. Sometimes I had to check and correct my own translations at least once to make them sound coherent in Spanish.
Vietnamese student	I could look at most of the sites and provided my changes. But, translation of technical sentences was tough. I think that the Vietnamese translations become much better and more natural now. ⁹
French student	This work session for the enhancement of TUT's website was a good idea and I am sure it will permit the University to be well known abroad and reveal its potentialities to students and partners who plan to come in Japan for studies or partnership.

⁹ This Vietnamese student provided his/her feedback originally in Japanese. This is the translation of the original feedback.

Our next step is to examine the accuracy and the quality of these crowd-sourced translations from the students. We are currently working on this together with professional translators, and would like to make a report on this investigation in the future.

Acknowledgments. This project was funded by International Affairs Division at Toyohashi University of Technology, and we would like to give special thanks to all the members of International Affairs Division for their support during the project. We are also thankful to Midori Tatsumi for her feedback on our paper.

References

1. Allen, J.: Post-editing. In: Somers, H. (ed.) *Computers and Translation: A Translator's Guide*, pp. 297–317. John Benjamins Publishing Company, Amsterdam (2003)
2. Allen, J.: What is post-editing? *Translation Automation* 4, 1–5 (2005)
3. O'Brien, S.: Methodologies for measuring the correlations between post-editing effort and machine translatability. *Machine Translation* 19(1), 37–58 (2005)
4. Guerberof, A.: Productivity and quality in MT post-editing. In: *MT Summit XII – Workshop: Beyond Translation Memories: New Tools for Translators MT*, Ottawa, Ontario, Canada, p. 8 (2009a)
5. Guerberof, A.: Productivity and quality in the post-editing of outputs from translation memories and machine translation. *Localisation Focus. The International Journal of Localisation* 7(1) (2009b)
6. Koehn, P., Haddow, B.: Interactive Assistance to Human Translators using Statistical Machine Translation Methods. In: *MT Summit XII* (2009)
7. Ambati, V., Vogel, S.: Can crowds build parallel corpora for machine translation systems? In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Los Angeles, CA, pp. 62–65 (2010)
8. Zaidan, O.F., Callison-Burch, C.: Crowdsourcing translation: professional quality from non-professionals. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 1220–1229 (2011)
9. Ambati, V., Vogel, S., Carbonell, J.: Active learning and crowd-sourcing for machine translation. *Language Resources and Evaluation (LREC)*
10. Callison-Burch, C.: Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In: *Proceeds of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 286–295 (2009)
11. Higgins, C., McGrath, E., Moretto, L.: MTurk crowdsourcing: A viable method for rapid discovery of Arabic nicknames? In: *NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*, Los Angeles, CA, pp. 89–92 (2010)
12. Yamamoto, K., Aikawa, T., Isahara, H.: Impact of collective intelligence on the post-editing machine translation output (機械翻訳出力の後編集の集合知による省力化). In: *Proceedings of NLP 2012, Hiroshima, Japan* (2012)

Translation of Quantifiers in Japanese-Chinese Machine Translation

Shaoyu Chen and Tadahiro Matsumoto

Department of Information Science, Gifu University
1-1 Yanagido, Gifu 501-1193, Japan

Abstract. In this paper we describe a translation method for numeral quantifiers in rule-based Japanese-Chinese machine translation. In Chinese, as in Japanese, numerals quantify nouns along with appropriate measure words (classifiers); numerals cannot quantify nouns by themselves (especially in Chinese). Those numeral quantifiers often cause errors in Japanese-Chinese machine translation because of the wide variety of the Chinese classifiers and complicated correspondence between Japanese and Chinese classifiers. It depends on the quantified noun which classifier is appropriate. Accordingly, we devised a method for classifying the quantified nouns by their semantic attributes through comparative analysis of quantifiers in Chinese and Japanese sentences, and obtained translation rules for quantifiers that use the classification method. We conducted an experiment on the translation rules with our experimental Japanese-Chinese MT system and by manual using 600 sentences. The translation rules achieved an accuracy of 92.8%, which indicates their effectiveness.

Keywords: quantifier, classifier, Chinese, Japanese, machine translation, measure word, counter word.

1 Introduction

A numeral, together with a measure word (classifier), quantifies a noun in Chinese and Japanese. Such numeral quantifiers often cause mistranslation in machine translation (MT) from Japanese to Chinese, although several practical MT systems have been available these days. The reasons for the difficulty are that: Chinese has many different kinds of classifiers; the relationship between the classifiers and the quantified nouns is not straightforward; the correspondence between Japanese and Chinese classifiers is also complicated.

Asahioka et al.[1] and Kamei et al.[2] presented classifications of Japanese quantifiers for natural language processing of Japanese. Hung[3] investigated quantifiers in translation corpus to find out the characteristics and difference of them in Japanese and Chinese, intending to help the learners' understanding of both languages. Yin et al.[4] proposed a method for translating quantifiers for rule-based Chinese-Japanese MT, which used semantic similarity between the nouns modified by quantifiers. However, there still are problems that need to be

further explored for Japanese-Chinese MT, which arise from the wider variety of Chinese classifiers and *quantifier floating* in Japanese.

While both languages has non-numeral quantifiers (e.g. *some*, *many* and *all*), we concentrate on numeral quantifiers in this paper.

1.1 Brief Summary of Quantifiers

Measure words, also called classifiers, are used along with numerals to define the quantity of given objects, or with demonstratives to identify specific objects. These words are called 助数詞 *jōsūshi* (counter)[5] in Japanese and 量词 *liangci* in Chinese.

Whenever a noun is preceded by a numeral or a demonstrative, a measure word must come in between them. The combination of numerals and measure words become quantifiers, which is named 数量詞 *shuliangci* in Chinese grammar. Phrases consisting of a number, a classifier, and a noun are known as “classifier phrases,” such as 一棵树 *yi-ke-shu* (one tree) and 两朵花 *liang-duo-hua* (two flowers).

1.2 Various Quantifiers in Chinese

Each quantifier in Chinese is associated with a noun, and a specific quantifier should be used before a specific noun, thus making Chinese more colorful and vivid.

Different classifiers often correspond to different particular nouns. Within categories there are further subdivisions. For example, while most animals take 只 *zhi*, domestic animals take 头 *tou*, long and flexible animals take 条 *tiao*, and horses take 匹 *pi*. Similarly, while long things that are flexible or soft take 条 *tiao*, long things that are stiff take 根 *gen*; however, if they are also round, such as pens or branches, they take 支 *zhi*. Classifiers also vary in how specific they are; 朵 *duo* is only used for flowers. Notwithstanding the same thing, classifiers vary in form. While 株 *zhu* is available for grass when it is seedling, 棵 *ke* is appropriate when it becomes adult[6].

Consequently, there is not a one-to-one relationship between nouns and classifiers; the same noun may be paired with different classifiers in different situations. In this research we classified the translation rules according to the characteristics of Chinese quantifiers.

2 Classifications of Quantifiers for Japanese-Chinese MT

There are various types of quantifier patterns in Japanese. By examining Japanese sentences with quantifiers chosen from KOTONOHA[10] corpus, we have summarized Japanese quantifier patterns as shown in Table 1.

Table 1. Patterns of quantifiers extracted from corpus.

Japanese sentence	quantifier pattern
^{THREE} 3つ ^{APPLE} の ^{リンゴ} ^を ^{ATE} 食べた。	Q+の(<i>no</i>)+N
^{APPLE} リンゴ ^{THREE} 3つ ^を ^{ATE} 食べた。	N+Q+を(<i>wo</i>)+V
^{APPLE} リンゴ ^は ^{THREE} 3つ ^だ 。	N+は(<i>ha</i>)+Q+だ(<i>da</i>)
^{APPLE} リンゴ ^を ^{THREE} 3つ ^{ATE} 食べた。	N+を(<i>wo</i>)+Q+V
^{THREE} 3つ ^{APPLE} リンゴ ^を ^{ATE} 食べた。	Q+N+を(<i>wo</i>)+V

2.1 Nominal Classifiers

According to Chinese grammar, quantifiers can be classified to two types: nominal classifiers and verbal classifiers. The nominal classifiers also contain three types, which are named classifiers proper, measure words, and measurement units. The classifiers proper, which is also called count-classifiers, are used for naming or counting a single count noun. The measure words, which is called mass-classifiers as well, can be used with multiple types of nouns, such as 套 *tao* (set) and 群 *qun* (group). The third type of nominal classifiers are the measure units (or measure items) such as 時間 *jikan* (hours), ヶ月 *kagetsu* (months), 年間 *nenkan* (years) and メートル (meters).

In Japanese, if a quantifier precedes to the quantified noun, の *no* (the genitive case particle) is required between the quantifier and the noun.

One-to-one. Some Japanese quantifiers can be uniquely translated into Chinese. We call this type of quantifiers “*one-to-one*.”

(Japanese)	(Chinese)	(English)
^{YEAR OLD} 3 歳 ^{CHILD} の ^{子供}	^{YEAR OLD} 3 岁 ^{CHILD} 的 ^{孩子}	a three-year-old child
^{YEARS} 5年間 ^{STUDY} の ^{勉強}	^{YEARS} 5年 ^{STUDY} 的 ^{学习}	a five-year study
^{DAYS} 3日間 ^{TRIP} の ^{旅行}	^{DAYS} 3天 ^{TRIP} 的 ^{旅行}	a three-day trip

One-to-many. However, because of the variety of Chinese quantifiers, Japanese quantifiers sometimes correspond to more than one Chinese quantifiers. Therefore, careful consideration is necessary to translate them correctly. We call this type of quantifiers “*one-to-many*.”

(Japanese)	(Chinese)	(English)
3本の { ^{BEER} ビール	3瓶 ^{BEER} 啤酒	three bottles of beer
ペン	3支 ^{PEN} 钢笔	three pens
バナナ	3根 ^{BANANA} 香蕉	three bananas

2.2 Verbal Classifiers

The other Japanese counter words, such as 回 *kai* and 度 *do*, are used to count actions and events, which should be translated as various Chinese measure words as shown in the following examples.

Japanese	Chinese	English
<small>JAPAN ONCE GO</small> 日本に一回行った	<small>GO TOKYO ONCE</small> 去了东京一趟。	have been to Japan once.
<small>THE PROGRAM TWICE WATCH</small> その番組を二回見たことがある	<small>THE PROGRAM WATCH TWICE</small> 那个节目看过两次。	have seen the program twice.
<small>THIS NEWS THREE TIMES HEAR</small> このニュースを三回聞いたことがある	<small>THIS NEWS HEAR THREE TIMES</small> 这个新闻听过三遍。	have heard the news three times.
<small>TARGET GUN FOUR TIMES SHOOT</small> ターゲットに銃で四回打った	<small>GUN TARGET HIT FOUR TIMES</small> 用枪对着目标打了四下。	hit goal with gun four times.

3 A Solution for Japanese-Chinese Translation of Quantifiers

We are developing an experimental Japanese-Chinese MT system *jaw/Chinese* on *jaw*[7], an MT engine from Japanese to other languages, which is based on a pattern transfer method. We have improved the translation rules for quantifier expressions, especially for the *one-to-many* problem.

3.1 Translation of Nominal Classifiers

One-to-one. First, *jaw/Chinese* analyzes the input Japanese sentence using IBUKI[8] (a Japanese morphological and *bunsetsu* (phrase) dependency analyzer), and generates the *bunsetsu* dependency tree, called InputTree (IT). Table 2 is the IT for the following Japanese phrase:

(ex.1) YEARS STUDY 3年間の勉強 (a three-year study)

Then IT is checked through three types of pattern rules (called Base, AdditionCW, and AdditionFW type¹) in the Japanese pattern dictionary. According to the role of 年間 *nenkan* (years) in the sentence, base type is suitable for this case. The tree built up with the rule patterns matching with an IT is called TransferTree (TT). Figure 1 illustrates the TT for the IT in Table 2.

¹ Base type: A case-frame-like rule and deals with the translation of basic propositional contents such as nouns and verbs.

AdditionCW, AdditionFW type: Mainly deal with adverbial expressions and conjunction expressions, which are optionally added to the Base type expressions such as AdditionCW とても *totemo* (very) and AdditionFW と *to* (and).

Table 2. Structure of InputTree (the output of IBUKI) for sentence (ex.1). (BID is the ID of the *bunsetsu* (or word), which depends on the *bunsetsu* specified by RBID. E1–E6 are function words in the *bunsetsu*. E7 is additional information about the dependency relation.)

BID	RBID	Category	Content word	E1	E2	E3	E4	E5	E6	E7
1	2	numeral	3							compound
2	3	suffix	年 <small>YEARS</small> 間				の			→(noun,pronoun)
3	0	noun	勉強 <small>STUDY</small>							end

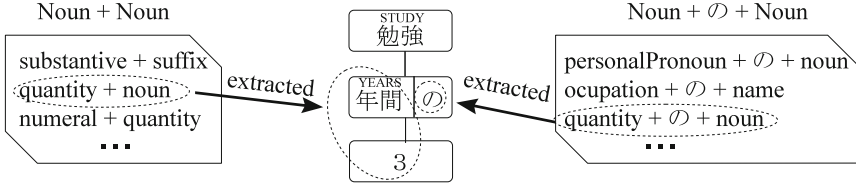


Fig. 1. Transfer Tree (TT) for “3年間の勉強” (a three-year study).

jav/Chinese is written in C#; the transfer rules are stored as C# methods in a DLL (dynamic link library). These methods construct a tree structure, called ExpressionTree (ET), whose nodes are C# objects corresponding to Chinese words. Each node has a *linearization* method, which generates its translation. By invoking the method of the root node of the ET, every node is invoked recursively to generate the output sentence. Finally, we can obtain the Chinese translation “3年的学习.”

One-to-many. However, if we use the translation rules introduced above, the problem will happen when we encounter *one-to-many* form. Because of the vast majority of classifiers in Chinese, the uniquely translated words maybe not exactly describe the objects when their properties are taken into consideration. For example, 本 *hon* is the Japanese classifier for long, thin objects. The Chinese equivalent for 本, however, varies with the objects.

Although each node object of ET has its Chinese translation in its `m_centerW` field, we omit that of the node for *one-to-many* quantifier (classifier) such as 本 *hon*. We defined `setUnit()` method of the `CNoun` class for nouns to determine the classifier for the noun. This method contains the classification information of quantifiers according to the semantic attribute of the noun. If the `m_centerW` field of the quantifier object is empty (i.e. *one-to-many*), the `setUnit()` method is invoked and returns the corresponding classifier as shown in Fig. 2.

The Necessity of Translation of 〇 (*no*). In Japanese, the genitive case particle 〇 (*no*) is often required between a nominal classifier and the noun, whereas its Chinese equivalent 的 *de* is not always required as shown in Table 3.

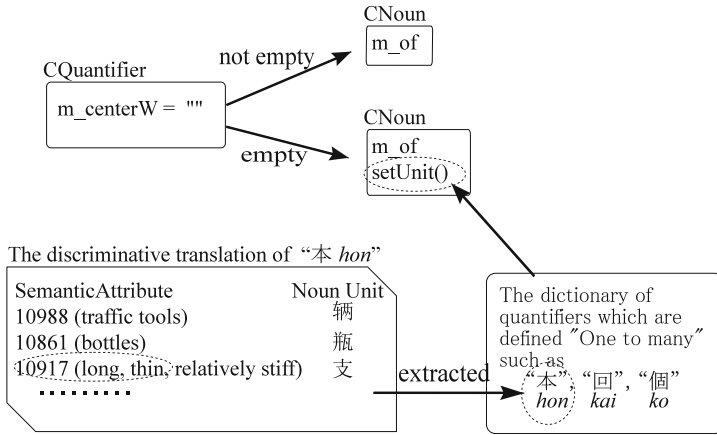


Fig. 2. Expression Tree (ET) for *one-to-many* classifier.

If the classifier is a count- or mass-classifier, 的 *de* is not necessary. Otherwise (that is, if the classifier is a measure unit) 的 *de* should be used to connect the quantifier and the noun.

Table 3. の (*no*) connecting quantifiers to nouns, and their Chinese translations.

Type of nominal classifiers	Translation of “の”	Japanese phrase	Chinese translation
count-classifier	ellipsis	A SHEET OF PAPER 1枚 <u>の</u> 紙	A SHEET OF PAPER 1张 纸
mass-classifier	ellipsis	SET OF BOOK 2セット <u>の</u> 書籍	SET BOOK 2套 书籍
measure units	necessary	METER ROAD 3メー <u>テ</u> ル <u>の</u> 道	METER ROAD 3米 <u>的</u> 路

3.2 Translation of Verbal Classifiers

The verbal classifiers are used for counting actions and events rather than objects as mentioned above. 回 *kai* and 度 *do* are Japanese, 遍 *bian*, 场 *chang*, 次 *ci*, 下 *xia*, 趟 *tang*, 声 *sheng*, 顿 *dun* and 回 *hui* are Chinese verbal classifiers. The sentence (ex.2) below is an example that has a verbal classifier, for which the result of Japanese analysis by IBUKI (i.e. InputTree) is shown in Table 4.

(ex.2) ^{AMERICA}アメリカに^{ONCE GO}1回行った。(Have been to America once)

In the sentence (ex.2), the quantifier “1回” (once) is adverbial; it modifies the action “アメリカに行く” (go to America). In *jaw*, AdditionCW type is used for writing pattern rules that deal with adverbial (and conjunctive) expressions,

Table 4. InputTree for sentence (ex.2).

BID	RBID	Category	Content word	E1	E2	E3	E4	E5	E6	E7
1	4	noun	^{AMERICA} アメリカ		^{TO} に					→(verb,adjective)
2	3	numeral	1							compound
3	4	suffix	^{TIMES} 回							→(verb,adjective)
4	0	predicate	^{GO} 行く	(PAST) た						end

so that we defined the translation rule for 回 *kai* as an AdditionCW type rule as shown in Fig. 3.

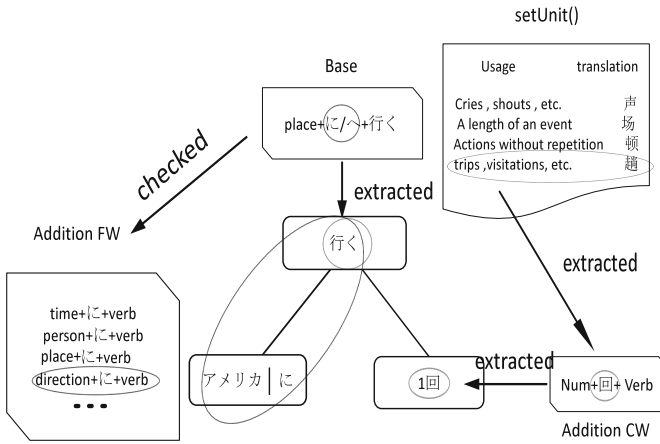


Fig. 3. Transfer Tree (TT) for verbal classifier.

Because of the limited number of Chinese verbal classifiers, jaw/Chinese can translate them more accurately in comparison with the *one-to-many* nominal classifiers.

3.3 Quantifier Floating

Japanese is an agglutinative language, while Chinese is an isolated language. It seems that quantifiers' positions in Japanese sentences are more flexible than in Chinese. For a Chinese sentence “买了一张桌子” (bought a table) with a quantifier, all of the five sentences in Table 5 are its Japanese equivalents.

Besides, if “noun+quantifier” is regarded as the subject or the direct object in a sentence, it results in more troublesome situation for matching rule patterns for quantifiers. To solve this problem, we defined a rule pattern using a virtual word “_” that can connect a noun with a quantifier as shown in Fig. 4.

Table 5. Japanese equivalents of “买了一张桌子” (bought a table).

Japanese sentences	Type of the quantifier in <i>jaw</i>
1 テーブル ^{TABLE} を1つ ^{BUY} 買った。	Addition CW
2 テーブル ^{TABLE} 1つ ^{BUY} を買った。	Base (Addition FW is necessary)
3 テーブル ^{TABLE} は1つ ^{BUY} 買った。	Addition CW
4 1つ ^{TABLE} テーブル ^{BUY} を買った。	Base (Addition FW is necessary)
5 1つの ^{TABLE} テーブル ^{BUY} を買った。	Base

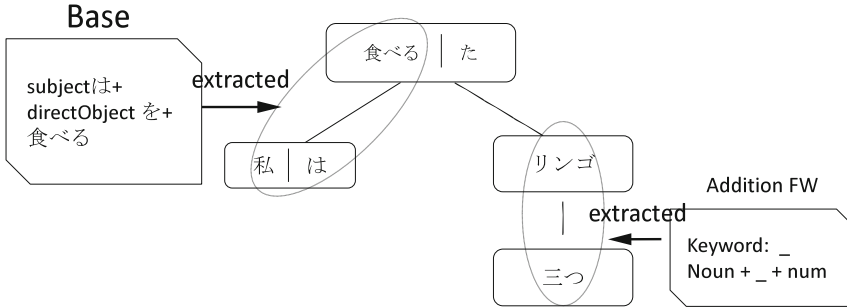


Fig. 4. Transfer Tree for quantifier floating.

The phenomenon that the quantifier moves away from the modified noun and modifies the verb is called *quantifier floating*. The attribute of the quantifier is adverb rather than adjective then. Accordingly, *jaw*/Chinese could not distinct whether the quantifiers modifies the noun or the verb in the sentence.

The following is an example of *quantifier floating* (“I ate three apples.” in English) and its translations.

Japanese	私はリンゴ ^{APPLE} を三つ ^{THREE} 食べた ^{EAT} 。
Current translation	我吃了苹果三个
Correct translation	我吃了三个苹果

Analysis: 三つ *mitsu* (three) modifies the object リンゴ *ringo* (apple) in our judgment. Nevertheless, the quantifier is placed on the end of the sentence by *jaw*/Chinese. The reason is that it was mistaken to be the verbal classifier.

Selection of Optimum Solution in *jaw*. There are three pattern types of translation rules in *jaw*, which are Base, AdditionCW and AdditionFW type. An input sentence (InputTree) is typically expressed with more than one combinations of those patterns, and therefore the best combination should be selected somehow. We have given the order of priority to the types. For example, Base patterns take priority over Addition types, and the combination that has less number of components (patterns) has higher priority[9].

4 Translation Experiment

We conducted an experiment on the translation rules of jaw/Chinese with 600 Japanese sentences (200 for each type of classifiers) that contain quantifiers. We made the sentences using KOTONOHA[10] (Balanced Corpus of Contemporary Written Japanese, BCCWJ), which is a Japanese corpus of 100 million words from books, newspapers, magazines, web and other documents. Because translation rule set of jaw/Chinese is still small, we limited *one-to-many* nominal classifiers to 本 *hon*, 回 *kai*, 人 *nin* and 枚 *mai* in order to complete the experiment with jaw/Chinese. For *one-to-one* and verbal classifiers, we judged accuracy of the rules by manual with various quantifiers. We evaluated the accuracy by finding out whether the translations of the quantifiers were appropriate and whether the words were in the correct order.

4.1 Experimental Results

The results of the experiment are summarized in Table 6. We compared the results with those of the OCN translation service² (hereinafter shortened to OCN), which is an online translation service employing machine translation technology of the commercial MT software J-Beijing.

Table 6. Experiment result.

Classifier type	Sentences	Correct		Accuracy rate	
		A	B	A	B
Nominal classifiers (one-to-many)	200	101	177	50%	88.5%
Nominal classifiers (one-to-one)	200	163	191	81.5%	95.5%
Verbal classifiers	200	155	189	77.5%	94.5%

(A: OCN translation service, B: jaw/Chinese)

In this section, the input Japanese sentences are numbered as (J-*n*), such as (J-1) and (J-2). The output Chinese sentences for (J-*n*) translated by OCN and by jaw/Chinese are numbered as (C-*na*) and (C-*nb*) respectively. The sentences numbered as (C-*nc*) are correct Chinese translations for (J-*n*).

One-to-many Nominal Classifiers. In this experiment, the translation accuracy of our rules is superior to that of OCN especially for *one-to-many* nominal classifiers. For example, the Japanese classifier 枚 *mai* in the sentence (J-1) below is used for thin, flat objects, such as papers, photographs and plates. Its typical Chinese equivalents are 张 *zhang* (for papers), 个 *ge* (for plates), 片 *pian* (for leaves), and 件 *jian* (for clothes). OCN translated 枚 *mai* as “张(件)” incorrectly as (C-1a). The appropriate Chinese classifier for handkerchiefs (blackboards, land, etc) is 块 *kuai*, which jaw/Chinese selected as (C-1b), judging from the quantified noun (i.e. handkerchief).

² <http://www.ocn.ne.jp/translation/>

(J-1) ハンカチ1枚は100円です。(One handkerchief is 100 yen.)

(C-1a) 手帕1张(件)是100日元。〈incorrect〉

(C-1b) 1块手帕100日元。〈correct〉

One-to-one Nominal Classifiers. The Chinese equivalent of the Japanese classifier ページ (page) in (J-2) is 页 *ye*, which was properly selected in both (C-2a) and (C-2b). The point in this case is the position of the quantifier 第20页 (the 20th page) in the sentences. OCN placed the quantifier before the quantified noun 书 *shu* (book) as (C-2a), whereas jaw/Chinese placed it after the verb phrase 读到了 (have read) correctly as (C-2b).

(J-2) 私はその本を第20ページまで読んだ。(I have read the book to the 20th page.)

(C-2a) 我到第20页读了那本书。〈incorrect in its position〉

(C-2b) 我那本书读到了第20页。〈correct〉

The Chinese equivalent of the Japanese quantifier “3人” *san-nin* (three people) is 三个人 *san-ge-ren*, which is found in the jaw/Chinese’s output (C-3b); and besides the word order of the quantifier and the quantified noun 学生 *xuesheng* (student) in (C-3b) seems to be correct. However, “学生3人” (three students) in (J-3) should be translated as 三个学生; that is, 人 *ren* (person) must be removed as (C-3c) because 学生 (student) conflicts with 人 (person).

(J-3) 彼は学生3人と一緒に行く。(He goes together with three students.)

(C-3a) 他与学生3人一起去。〈incorrect〉

(C-3b) 他和三个人学生一起去。〈incorrect〉

(C-3c) 他和三个学生一起去。〈Correct translation〉

Verbal Classifiers. The Japanese classifier 度 *do* has two different roles: a verbal classifier and a measure unit. This fact made difficult for jaw/Chinese to select appropriate classifiers.

In (J-4), 度 *do* is a verbal classifier, which refer to the number of times of the action “クリックする” *kurikku-suru* (click). This 度 *do* was correctly translated as 次 *ci* (C-4b).

(J-4) もう一度クリックすると、選択を解除できる。
(If you click it again, you can cancel the choice.)

(C-4b) 再点击一次的话, 就能解除选择。〈correct〉

度 *do* in (J-5) was also translated as 次 *ci* as (C-5b). However, in this case 度 *do* is a measure unit (degree), and therefore 度 *du* in (C-5c) was the correct translation.

(J-5) 午後1時過ぎに35度を観測した。

(A temperature of 35 degrees was observed a little after 1 pm.)

(C-5b) 下午1点过后，观测了35次。〈incorrect〉

(C-5c) 下午1点过后，观测到35度。〈Correct translation〉

4.2 Other Problems

Separation of Quantifiers from the Modified Noun. In the sentence (J-6) below, the quantifier “1本” *ippon* and the modified noun 草 *kusa* (grass) are separated in different clauses. This separation led to mistranslation for the reason that jaw/Chinese failed to find the modified noun corresponding to the quantifier “1本” *ippon* in the sentence.

(J-6) ^{THIS}この^{AREA}地域の^{GRASS}草は、^{ENVIRONMENT}環境が^{AGGRAVATED}悪化して^{BECAUSE}いたため、^{GROW}1本も^{NOT}生えててこない。

(No blade of grass grows in this area because the environment has been aggravated.)

Multiple Modified Nouns. In (J-7) the quantifier “1本” *ippon* quantifies three nouns, which should be translated as 瓶 *ping* (for beer), 根 *gen* (for rolled sushi) and 个 *ge* (for ice cream). Our translation rules, however, does not work well to solve this problem.

(J-7) ^{DRAFT BEER}生ビール、^{ROLLED SUSHI}細巻寿司、^{ICE CREAM}アイスクリーム、^{ONE}1本^{GIVE}ずつください。

(I'll have a bottle of beer, a rolled sushi, and an ice cream.)

Moreover, the influence of *quantifier floating*, which brought about several mistakes, reduced the accuracy rate in the experiment.

5 Conclusions

We analyzed the usage and differences of quantifiers in Chinese and Japanese in order to solve the problems of mistranslation of various quantifiers in Japanese-Chinese machine translation. We proposed translation rules which classify the classifiers successfully in three types, which are called *one-to-one*, *one-to-many* and verbal classifiers. Our translation rules were achieved the accuracies of about 92.8% in a small- scale experiment in our *jaw* MT system.

However, there remains some issues that need to be solved, such as quantifiers that are separated from the modified nouns in different clauses, and existence of more than one nouns modified by a single quantifier. Furthermore, while our

discussion about translation methods was focused on sentences in which a single quantifier exists, compound quantifiers (multiple quantifiers in a sentence) will also cause problems. For instance, it must be solved how to decide the positions of the quantifiers in the translation of the following sentence: “1^{A DAY}日 1^{ONCE}回 3^{THREE}つ 食^{EAT}べる。”

References

1. Asahioka, Y., Hirakawa, H., Amano, S.: A semantic classification of Japanese numerical expressions. In: Proceedings of the 40th National Convention IPSJ, pp. 470–471 (1990)
2. Kamei, K., Muraki, K.: Analysis of Japanese counter words. In: Proceedings of the 41th National Convention IPSJ, pp. 155–156 (1990) (in Japanese)
3. Hung, Y.: A study of the uses of quantifiers in translation corpus. IEICE Technical Report, TL106(363), 37–42 (2006)
4. Yin, D., Shao, M., Jiang, P., Ren, F., Kuroiwa, S.: Treatment of Quantifiers in Chinese-Japanese Machine Translation. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS (LNAI), vol. 4114, pp. 930–935. Springer, Heidelberg (2006)
5. Wikipedia: Japanese counter word,
http://en.wikipedia.org/wiki/Japanese_counter_word
6. Wikipedia: Chinese classifier,
http://en.wikipedia.org/wiki/Chinese_classifier
7. Ikeda, T.: The struggle to develop machine translation systems from Japanese to Asian languages. *Japanese Linguistics* 28(12), 62–71 (2009) (in Japanese)
8. Ikeda, T., Wakita, T., Oguchi, T.: Japanese bunsetsu analyzer ibukiC (v0.20) and the introduction of functional bunsetsu into it. In: Proceedings of the 14th Annual Meeting of the Association for Natural Language Processing, pp. 221–224 (2008) (in Japanese)
9. Yiou, W.: The Study on Japanese-Chinese Machine Translation System — Focusing on the Translation of the Existential Expression and the Light Verb Construction. PhD dissertation, Gifu University (2006)
10. National Institute for Japanese Language and Linguistics: KOTONOHA,
<http://www.ninjal.ac.jp/english/products/kotonoha/>

Toward Practical Use of Machine Translation

Hitoshi Isahara

Toyohashi University of Technology
isahara@tut.jp

Abstract. This paper presents an overview of our research activities on machine translation conducted at Toyohashi University of Technology. I focus on how to make machine translation useful for real world business, not mentioning quality improvement of translation engines. I present here three approaches for making machine translation practical; i.e. simplifying the Japanese source text, extracting and listing salient expressions and their equivalents in a document and enhancing the post-editing process. This study is important from both a business perspective and an academic perspective.

Keywords: Machine Translation, Simplified Language, Term Extraction, Post-editing.

1 Introduction

Various services, such as information retrieval and information extraction, using natural language processing technologies trained by huge corpora have become available. In the field of machine translation (MT), corpus-based machine translations, such as statistical machine translation (SMT) and example-based machine translation (EBMT), are typical applications of using such volumes of data in real business situations. Thanks to such huge available data, current machine translation system is enough high quality for some specific language pairs. But still some people have doubt about usefulness of machine translation, especially for translation among different types of languages, such as Japanese and English. One study examined for what types of people current machine translation systems are useful [1], by simulating the retrieval and reading of web pages in a language different from one's mother tongue. However, there has been little research to verify the technologies which make MT systems more useful in real world situations.

In this paper we focus on how to make machine translation useful for real world business. I present here three approaches; i.e. simplifying the Japanese source text, extracting and listing salient expressions and their equivalents in a document and enhancing the post-editing process. This study is important from both a business perspective and an academic perspective.

2 Problems of Translation between Japanese and English

Developers of Japanese-to-English and English-to-Japanese machine translation systems face more difficulties than counterparts providing systems translating, for example, English-to-French. This is because Japanese is very different in syntax and semantics from English, so we often need some context to translate Japanese into English (and English into Japanese) accurately. English uses a subject-verb-object word order, while in Japanese, the verb comes at the end of the sentence, i.e. a subject-object-verb order. This means that we have to provide much more example sentence pairs of Japanese and English compared to when translating most European languages into English, as they also use a subject-verb-object order. The computational power required for Japanese to come up with accurate matches is enormous. And accuracy is particularly necessary for businesses selling their products overseas, which is the reason why it is needed to help Japanese companies provide better translated manuals for their products.

Faced with such obstacles, we conduct researches on quality improvement of MT engines, which include five-year national project on development of Japanese-Chinese machine translation system [2]. In parallel with this kind of MT research, we are taking a three-step approach to improve the MT quality in real life environment: simplifying the Japanese source text (controlled language), enriching lexicon and enhancing the post-editing process (Figure 1).

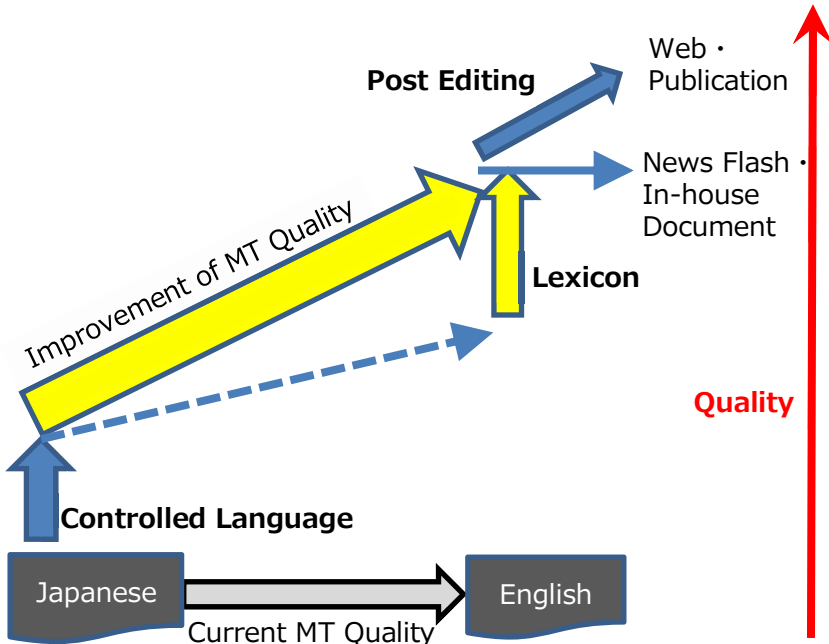


Fig. 1. Quality of Translation

For controlled language, e.g., simplified Japanese, we are devising a set of guidelines and rules for writers of Japanese manuals that will be used as the source for MT. These rules include writing shorter and simpler sentences; adding the subject when missing; and providing context when there is ambiguity. As for enriching lexicon, we are extracting and listing salient expressions and their equivalents in a document and store them into translation dictionary. For post-editing, which can be costly and time-consuming, we are conducting an experiment using foreign students of Toyohashi University of Technology to post-edit the MT output of English version of the university's Web site into their own languages.

Combining above mentioned techniques, MT will be practically useful tools for real world translation. Our approach, then, is not to focus on just one aspect of MT. Rather we want to improve and support the entire machine translation process.

3 Simplified Japanese [3,4]

Output quality of MT heavily depends on the quality of analysis of input sentences. Long and complex sentences in syntax and semantics are mostly very difficult for automatic analyzers to output proper structures. Therefore, restricting structures of input text will be beneficial for MT system to achieve high quality translation. We seek to address this challenge by investigating the feasibility of developing a 'controlled Japanese' with explicit restrictions on vocabulary, syntax and style adequate for authoring technical documentation. This research is being conducting in collaboration with an automobile related company in Japan.

We aimed to build translation awareness within a global Japanese company where non-professional authors are called upon to write 'global job manuals' for internal dissemination. Following an analysis of current practice, we devised a document template and simple writing rules which we tested experimentally with MT systems. Sentences violating the rules were extracted from the original data and rewritten in accordance with the respective rule. The original and rewritten sentences were then translated by MT systems, and the inputs and outputs were submitted to human evaluation. Overall, native-speaker judges found that the quality of the Japanese was maintained or improved, while the impact on the raw English translations varied according to MT system. Then, we explained our template and rules to employees of the company and asked them to write their manuals articulating the know-how using our template and rules. We are currently investigating their documents and trying to identify the most promising avenues for further development.

One of the other possibilities of controlled (or simplified) language is translation between two languages both of which are properly controlled. If we train SMT with parallel controlled language corpus, the SMT can translate controlled input into controlled output with high quality. Some of multilingual MT systems are combination of MT engines for two languages and translations between non-English languages are performed via English. Such cascade translation usually amplifies errors during translation. Using controlled English as a pivot would be promising solution of this problem.

4 Extracting and Listing Salient Expressions and Their Equivalents in a Document [5]

Quality of translation, especially its informativeness for human readers, is affected by whether technical expressions are translated properly or not. We are trying to compile automatically parallel term dictionary using documents in a specific domain.

There are several methods to acquire new words from large amount of text and some of them showed high performance for compound nouns. Our aim is to acquire technical terms which include not only compound nouns but also longer phrases such as “Extraction of Informative Expressions from Domain Specific Documents” in Japanese. The method uses morpheme based n-gram to save processing time and space, therefore the acquired terms are compounds of one or plural number of morphemes.

Because we use morpheme strings as input, each morpheme in the string is usually not an unknown words but stored in the dictionary of morphological analyzer. We extract compound nouns and longer phrases which are new terms as a whole. Or, we can say we extract salient terms including compound nouns and noun phrases which are written in Japanese but may contain many English words.

Our term acquisition method consists of two stages: an extraction of candidate terms (“Candidate Selection”) and a guess as to terms (“Unithood Checking”). First, the statistical indicators we defined are used to select all one-morpheme to ten-morpheme strings that appear at least once in a large number of documents, and also appear repeatedly in several documents. In this way, we have enabled a computer to emulate a human sense to recognize and understand unknown terms. Next, the strength of connection between the constituent morphemes of each candidate term is assessed to arrive at a guess as to whether or not it is in fact a term.

We extracted salient terms both in Japanese and English from parallel documents, e.g., maintenance manuals for automobile, in collaboration with a translation company in Japan. The result is promising. We could extract salient phrases which contain 80% of terms expected by the company. Currently, we try to get equivalents of extracted terms using SMT trained with the parallel documents.

5 Crowdsourcing Post-editing [6]

With properly controlled input sentences and substantial dictionary, state of the art MT system is useful, for example, for quick translations, such as news flash, and in-house translations. (Figure 1)

For document which needs higher quality, post-editing is required. However, post-editing can be costly and time-consuming, and not everybody can pay for it. We are conducting a preliminary investigation on the impact of crowdsourcing post-editing through the so-called “Collaborative Translation Framework” (CTF) developed by the Machine Translation team at Microsoft Research. Crowdsourcing translation is an increasingly popular-trend in the MT community, and we hope that our approach can shed new light on the research into crowdsourcing translation.

For our project, we used foreign students at Toyohashi University of Technology (TUT) and asked them to post-edit the MT output of TUT's websites (<http://www.tut.ac.jp/english/introduction/>) via Microsoft Translator into their own languages using the CTF functionalities. Though we do not expect that students have the same degree of accuracy from the professionals, we can note that they have a better understanding of the context, and so this kind of collaboration could improve and reduce the cost of the post-editing process.

We finished first experiment using 22 foreign students attending our university to post-edit the MT output of English version of the university's Web site into their own languages. Currently, we are conducting an experiment using 4 Japanese students with more precise settings, such as ordering of post-editing.

6 Concluding Remarks

In this paper, we provided a brief overview of current research activities on machine translation, all of which are aimed to make machine translation practically useable. Though our research is not completed, some of the result obtained so far are promising.

Acknowledgement. This work was funded by the Strategic Information and Communication R&D Promotion Programme of the Ministry of Internal Affairs and Communications, Japan.

References

1. Fuji, M., et al.: Evaluation Method for Determining Groups of Users Who Find MT Useful. In: Proceedings of the Machine Translation Summit VIII (2001)
2. Isahara, H., et al.: Development of a Japanese-Chinese machine translation system. In: Proceedings of MT Summit XI (2007)
3. Tatsumi, M., et al.: Building Translation Awareness in Occasional Authors: A User Case from Japan. In: Proceedings of EAMT 2012 (2012)
4. Hartley, A., et al.: Readability and Translatability Judgments for "Controlled Japanese". In: Proceedings of EAMT 2012 (2012)
5. Yamamoto, E., et al.: Extraction of Informative Expressions from Domain Specific Documents. In: Proceedings of LREC 2008 (2008)
6. Aikawa, T., et al.: The Impact of Crowdsourcing Post-editing with the CollaborativeTranslation Framework. In: Proceedings of JapTAL 2012 (2012)

Phrase-Level Pattern-Based Machine Translation Based on Analogical Mapping Method

Jun Sakata, Masato Tokuhisa, and Jin'ichi Murakami

Information and Knowledge Engineering, Tottori University
4-101 Koyama-Minami, Tottori 680-8550, Japan
{d112004,tokuhisa,murakami}@ike.tottori-u.ac.jp

Abstract. To overcome the conventional method based on Compositional Semantics, the Analogical Mapping Method was developed. Implementing this method requires a translation method based on sentence patterns. Although a word-level pattern-based translation system already exists in Japanese-English machine translation, this paper describes the new phrase-level pattern-based translation system. The results of translation experiments show that the quality of phrase translation is still low. However, these problems are to be resolved in our future work.

Keywords: Pattern-Based Machine Translation, Sentence and Phrase Pattern, Phrase Translation.

1 Introduction

The conventional machine translation method based on compositional semantics has a problem in that it cannot generate sentence meaning when it generates the target sentence. To resolve this problem, Ikehara et al. proposed a transfer method named the “Analogical Mapping Method” [1]. This method uses sentence patterns that have “linear” and “non-linear” parts. In translation, we perform local translation of variables that are equivalent to the “linear part” and insert these results into the target sentence pattern.

Machine translation based on the Analogical Mapping Method requires many sentence patterns, a pattern matching system, and the translation system with matching sentence patterns. To implement Japanese-English MT based on the Analogical Mapping Method, we developed the “Japanese-English compound and complex sentence pattern dictionary”, a structural pattern matching system named “SPM” [2], and a generating system named “ITM”¹. The dictionary has 226,817 sentence pattern pairs from Japanese/English compound/complex sentences pairs. It also has three levels of sentence pattern: word-level (121,904 pattern pairs), phrase-level (79,438 pattern pairs), and clause-level (25,475 pattern pairs) [1]. The sentences were collected from various Japanese-to-English parallel corpora.

¹ “ITM” has never been described in a paper in detail.

Currently, only the word-level pattern-based translation is implemented in ITM. However, this pattern matching rate (SPM + ITM) is still low. To increase the pattern matching rate, we try to use phrase-level patterns [3]. For this, we need phrase translation. The phrase pattern dictionary has been developed from the Japanese-English compound and complex sentence pattern dictionary. Moreover, we implement the phrase-level pattern-based translation that performs phrase translation using this phrase pattern dictionary.

The rest of this paper is organized as follows. Section 2 describes an example of our sentence pattern dictionary and phrase pattern dictionary. Section 3 explains the ordinary word-level pattern-based translation method. Section 4 presents our proposed method. Section 5 discusses results of the experiments and assesses the proposed method. Finally, section 6 offers our conclusions.

2 Sentence Pattern Dictionary and Phrase Pattern Dictionary

2.1 Sentence Pattern

Fig. 1 shows an example of our sentence patterns. Sentence patterns have letters, variables, functions, and markers. Japanese/English word/phrase/clause alignment are replaced with variables. Word-level patterns have word variables. “ $N2^{\wedge}poss$ ” means it $N2$ is the possessive case in English. “ $V5^{\wedge}past$ ” means $V5$ is the past tense. In Japanese, a subject is often omitted, so “ $\langle N1 \text{ は } \rangle$ ” means whether the subject is omitted or not in pattern. In English patterns, $\langle I|N1 \rangle$ is “ $N1$ ” if Japanese matches $N1$, or “ I ” if not. “.hitei” and “.kako” are tense and modality function, respectively [4]. In total, 226,817 sentence patterns have already been created [1].

AC000004-00	
Japanese Sentence:	彼のお母さんがああ若いとは思わなかった。 [Kare no okaasan ga aa wakai towa omowa naka tta.]
English Sentence:	I never expected his mother to be so young.
Word-Level JP.Pattern:	$\langle N1 \text{ は } \rangle N2 \text{ の } N3 \text{ がああ } AJ4 \text{ とは } V5.hitei.kako.$
Word-Level EN.Pattern:	$\langle I N1 \rangle \text{ never } V5^{\wedge}past N2^{\wedge}poss N3 \text{ to be so } AJ4.$
Phrase-Level JP.Pattern:	$\langle N1 \text{ は } \rangle NP2 \text{ がああ } AJ3 \text{ とは } V4.hitei.kako.$
Phrase-Level EN.Pattern:	$\langle I N1 \rangle \text{ never } V4^{\wedge}past NP2 \text{ to be so } AJ3.$

Fig. 1. Description of Japanese-English compound and complex sentence pattern dictionary

2.2 Phrase Pattern

Our phrase pattern dictionary is automatically extracted from the Japanese-English compound and complex sentence pattern dictionary. Phrase patterns are extracted from the word-level sentence patterns.

Phrase patterns are categorized as noun phrases (*NP*), verb phrases (*VP*), adjective phrases (*AJP*), adjective-verb phrases (*AJVP*), and adverbial phrases (*ADVP*).

Fig. 2 shows examples of each phrase pattern. In the *VP* patterns, verbs are letters and not changed variables, because verbs have nonlinearity.

<NP>	
Japanese Pattern:	$N1 \text{ の } N2$ [$N1 \text{ no } N2$]
English Pattern:	$N1^{\wedge} \text{ poss } N2$
Original Japanese:	彼のお母さん [kare no okaasan]
Original English:	his mother
<VP>	
Japanese Pattern:	ああいう $N1$ と'付き合う' [aayuu $N1$ to 'tukiau']
English Pattern:	'associate' with that kind of $N1$
Original Japanese:	ああいう人と付き合う [aayuu hito to tukiau]
Original English:	associate with that kind of person
<AJP>	
Japanese Pattern:	実に $AJ1^{\wedge} \text{ rentai}$ [jituni $AJ1^{\wedge} \text{ rentai}$]
English Pattern:	very $AJ1$
Original Japanese:	実に痛ましい [jituni itamashii]
Original English:	very painful
<AJVP>	
Japanese Pattern:	$ADV1 \text{ } AJV2^{\wedge} \text{ rentai}$
English Pattern:	$ADV1 \text{ } AJ2$
Original Japanese:	とても静かな [totemo shizukana]
Original English:	very quiet
<ADVP>	
Japanese Pattern:	$N1 \text{ の } \text{あと}$ [$N1 \text{ no } \text{ato}$]
English Pattern:	after $N1$
Original Japanese:	手術のあと [shujutu no ato]
Original English:	after the operation

Fig. 2. Description of phrase patterns

3 Pattern-Based Machine Translation

3.1 Japanese Sentence Pattern Matching (SPM)

The Japanese pattern matching system named SPM has already been developed. The SPM [2] implements the Augmented Transition Network (ATN) algorithm

[5] with breadth-first search and uses sentence patterns. The input sentence for SPM is already morphological and has semantic codes added. SPM performs pattern matching between the input sentence and sentence patterns. Moreover, SPM outputs the pattern matching results. Fig. 3 shows an example of an input sentence.

-
1. /彼 (1710,{NI:23,NI:48})
 2. +の (7410)
 3. /お母さん (1100,{NI:80,NI:49})
 4. +が (7410)
 5. /ああ (1110)
 6. /若い (3106,{NY:5})
 7. +と (7420)
 8. +は (7530)
 9. /思わ (2392, 思う, 思わ,{NY:32,NY:31})
 10. +なかつ (7184, ない, なかつ)
 11. +た (7216)
 12. +。 (0110)
 13. /nil
-

Fig. 3. Example of an input sentence

In the first line in Fig. 3, “彼” is a Japanese morpheme, “1710” is tagging code, and “NI:23,NI:48” are indeclinable semantic codes [6]. In the sixth line, “NY:5” is a declinable semantic code. Each line shows a Japanese morpheme and semantic information.

PATTERN=PJAC000004-00
 =[NP2, が, ああ,AJ3, とは, V4,.hitei,.kako,。]
 =[1,2,3,4,5,6,7,8,9,10,11,12]=12
 NP2=[1,2,3]=3
 AJ3=[6]=1
 V4=[9]=1

Fig. 4. Results of sentence pattern matching

Fig. 4 shows the sentence pattern matching result. “AC000004-00” is the Japanese pattern ID. “NP2=[1,2,3]” shows that the morpheme numbers 1, 2, and 3 are matched as the phrase variable NP2. NP2 is “彼のお母さん [kare no okaasan]”. “AJ3=[6]” shows “若い [wakai]” is matched as AJ3. “V4=[9]” shows “思わ [omowa]” is matched as V4.

If several sentence patterns are matched, we select only one sentence pattern in accordance with the following steps.

1. Sentence patterns are selected by using semantic codes in the input sentence.

2. Pattern matching test is done with all Japanese sentences in the Japanese-English compound and complex sentence pattern dictionary.
The most matched pattern is selected.
3. If several patterns are left, only one pattern is randomly selected.

3.2 Word-Level Pattern-Based Machine Translation (ITM-w)

An English sentence is translated with matched sentence pattern pairs and the translation system “ITM”. ITM-w (word-level translation) performs word translation for Japanese linear parts of SPM results. Word translation is performed by a word dictionary. Several results of word translation are inserted into the English pattern, and the maximum likelihood sentence is selected by the English language model. The translation steps of ITM-w are as follows.

1. The sentence pattern pair is selected (section 3.1).
2. Word translations of Japanese linear parts are performed.
3. Word translation results are changed to the assigned form.
4. Candidates of word translation are inserted into the English sentence pattern.
5. The maximum likelihood sentence is selected by the English language model.

4 Phrase-Level Pattern-Based Machine Translation

We implemented phrase-level pattern-based translation. The proposed method is constructed of SPM-s (sentence pattern matching), SPM-p (phrase pattern matching), ITM-p (phrase-level translation), and ITM-w. SPM-s is already described section 3.1, and ITM-w is described section 3.2. SPM-p is the same program as SPM-s but uses phrase patterns. ITM-p performs phrase translation with phrase patterns and word translation with word dictionary. In ITM-p, SPM-p and ITM-w are activated for phrase translation.

Phrase patterns have word variables and are translated by the word dictionary. Word translation results are inserted into the phrase pattern, and the maximum likelihood phrase is selected by the English language model. If several phrase patterns are matched, all phrase patterns are used.

The steps in our proposed method are as follows.

Process 1 Sentence pattern matching is performed by SPM-s.

Process 2 The sentence pattern pair is selected (section 3.1).

Process 3 Phrase translation is performed in ITM-p.

Process 4 Phrase pattern matching is performed by SPM-p.

Process 5 Phrase translation is performed by ITM-w.

Process 6 Word translation is performed in ITM-p.

Process 7 Candidates for all local translation results are inserted into the English sentence pattern.

Process 8 The maximum likelihood sentence is selected by the English language model.

Fig. 5 sketches the whole configuration of phrase-level pattern-based translation.

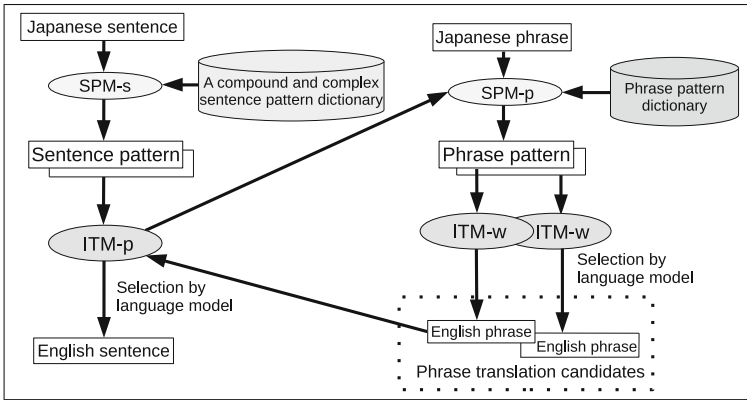


Fig. 5. Phrase-level ITM

4.1 Example of Phrase-level Translation (ITM-p)

Fig. 6 shows an example of translation.

Input Sentence:	彼のお母さんがああ若いとは思わなかった。 [kare no okaasan ga aa wakai towa omowa naka tta。]
Reference Sentence:	I never expected his mother to be so young.
English Pattern:	<I N1> never V4^past NP2 to be so AJ3 .
Output Sentence:	I never expected his mother to be so young .

Fig. 6. Example of translation results

Processes 1 and 2 are already described in section 3.1.

Process 3. In SPM-p, “彼のお母さん [kare no okaasan]” is matched as the phrase variable *NP2* and is translated by ITM-w. Phrase translation results (“his mother”, “he’s mother”,) are obtained with all phrase patterns.

Process 6. The Japanese morphemes “若い [wakai]” and “思わ [omowa]” are matched as the word variables *AJ3* and *V4*, respectively. They are translated by the word dictionary, and several candidates are obtained. “*V4^past*” means the past tense is selected from the translation candidates. For example, only past tense words (“thought”, “expected”,) are selected from translation candidates (“think”, “thought”, “expect”, “expected”,).

Process 7. All local translation results are inserted into the English pattern.

Process 8. The maximum likelihood sentence is selected by tri-gram. Fig. 7 shows an example of words selection by tri-gram.

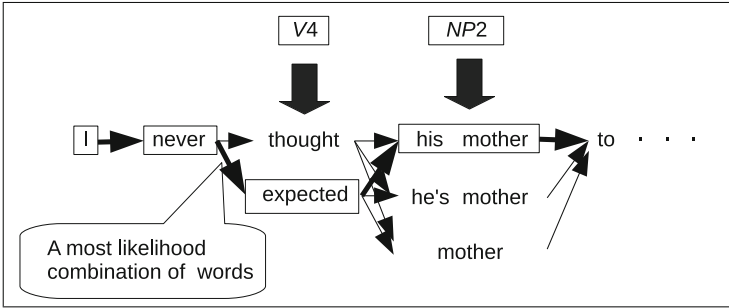


Fig. 7. Selection from translation candidates

5 Experiments

First, we carried out the closed-test to check the implementation of ITM-p. Next, we carried out the open-test to investigate effectiveness of the proposed method.

5.1 Closed-Test

Experimental Method. To survey our system, the closed-test was carried out with input sentences from the Japanese-English compound and complex sentence pattern dictionary. We used 500 input sentences for pattern matching and selected 100 sentences to evaluate translation accuracy. For pattern matching, we used 100 sentences that included at least one *NP* in the self-pattern. *VP*, *AJP*, *AJVP*, and *ADVP* were tested in the same way. We calculated the pattern matching rate and assessed the translation accuracy by human evaluation.

Pattern Matching Results. Table 1 shows the number of sentences that matched sentence patterns. In table 1, “Pattern mismatch” means that no pattern is matched to the input sentence. “Self-pattern mismatch” means that the self-pattern is not matched but other patterns are. “Self-pattern match” means that the self-pattern is matched.

Table 1. Results of self-pattern matching

	Pattern mismatch	Self-pattern mismatch	Self-pattern match
#Sentence <i>NP</i>	4	7	89
#Sentence <i>VP</i>	4	2	94
#Sentence <i>AJP</i>	7	4	89
#Sentence <i>AJVP</i>	11	16	73
#Sentence <i>ADVP</i>	20	27	53

Example of Pattern Mismatch. Fig. 8 shows examples of pattern mismatch and self-pattern mismatch.

<Pattern mismatch>	
Input St.	早くも子供のころからたくさん本を読むという傾向を示した。 [Hayakumo kodomo no koro kara takusan hon wo yomu to yuu keikou wo shimeshi ta.]
Self-Pt..JP.	<N1 は >ADVP2!VP3(という と言う)N4 を V5.kako. [<N1wa>ADVP2!VP3(toyuu toyuu)N4 wo V5.kako.]
<Self-pattern mismatch>	
Input St.	その品は品質がいいので高価なものももっともである。 [Sono shina wa hinshitu ga ii node koukana no mo mottomo dearu.]
Self-Pt..JP.	NP1 は N2 が AJ3^rentai ので AJV4^rentai!のも!VP5.tearu. [NP1 wa N2 ga AJ3^rentai node AJV4^rentai! nomo !VP5.tearu.]

Fig. 8. Examples of pattern mismatch and self-pattern mismatch

The following three reasons cause sentence pattern mismatching and self-pattern mismatching.

Cause 1 Sub-networks are short for SPM-s.

Cause 2 The sentence pattern description is wrong.

Cause 3 The result of morphological analysis is wrong.

Cause 1. “Pattern mismatch” in Fig. 8 is caused by a shortage of sub-networks for SPM-s. In this example, “早くも子供のころから [hayakumo kodomo no koro kara]” is *ADVP*. Sub-networks for *ADVP* are short, and this sentence is not matched to this pattern. Therefore, this sentence is not matched to any sentence patterns. In table 1, the number of pattern mismatches in *ADVP* is the largest. In *ADVP*, many cases of pattern mismatching are caused by this problem. They seem to require many sub-networks for each different structure to respective phrase patterns. To resolve this problem, we have to add sub-networks to SPM-s.

Cause 2. “Self-pattern mismatch” in Fig. 8 is caused by the wrong pattern. “もっともである [mottomo dearu]” is *AJVP*, but it is described as “*VP5.tearu*” in this sentence pattern. To resolve this problem, we need to examine and classify different kinds of wrong descriptions and correct them manually.

Cause 3. We omitted the case for failing morphological analysis.

Human Evaluation. Table 2 shows evaluation criteria.

Table 2. Evaluation criteria

Eval. 1	The sentence structure is correctly composed, and all local translation are not mistranslation.
Eval. 2	The sentence structure is correctly composed, but a local translation is mistranslation.
Eval. 3	The sentence structure is incorrectly composed.

Table 3 shows the evaluation results.

Table 3. Evaluation results

	Eval.1	Eval.2	Eval.3
#Sentence	27	68	5

Fig. 9 shows examples of Eval. 1 and Eval. 2. All results in Eval. 3 are caused by wrong sentence pattern description. In Eval. 1, the sentence structure and all local translations are correct. In Eval. 2, sentence structure is correct, but local translations of *NP1* and *N3* are not correct.

<Eval.1>

Input Sentence:	これを読んで泣かざるを得ぬ。 [Kore wo yon de naka zaru wo e nu.]
Reference Sentence:	I can not read this without crying.
English Pattern:	<I N1> can not VP2^base without V3^ing.
Matched Morphemes:	VP2 = kore wo yon, V3 = naka
Output Sentence:	I can not read this without crying.

<Eval.2>

Input Sentence:	その晚餐会は彼をたたえるために開かれた。 [Sono bansankai wa kare wo tatae ru tameni hiraka re ta.]
Reference Sentence:	The dinner party was a tribute paid to him.
English Pattern:	NP1 @be^past N3 paid to N2^obj .
Matched Morphemes:	NP1 = Sono bansankai, N2 = kare, N3 = tatae
Output Sentence:	That it was name paid to him .

Fig. 9. Examples of Eval.1 and Eval.2

In Table 3, the translation accuracy is low in spite of the self-sentence pattern being used. The most significant reason for this is the selection of the improper phrase translation. Such an example is shown in Eval. 2 in Fig. 9. “その晚餐会 [sono bansankai]”, which corresponds to “the dinner party”, is matched

as *NP1*, and the selected translation result is “that it”. In this case, selected phrase translation uses the phrase pattern “*AJ1 N2^pron*”. “*その* [sono]” is matched as *AJ1*, and the translation result of “*その* [sono]” is “that”. “*晩餐会* [bansankai]” is matched as *N2^pron*. “*N2^pron*” means that *N2* is the pronoun. Thus, the translation result of “*晩餐会* [bansankai]” is changed to the pronoun “it”. On the other hand, the correct local translation “the dinner party (*AJ1 N2*)” is included in the candidates. This suggests failed selection from sentence candidates in ITM-p. It seems that the selection by the language-model is not good enough.

Functions in phrase patterns were added in sentence patterns. They are often improper in phrase patterns and often cause improper phrase translation results. If improper pronouns are generated by the functions in phrase translation, these phrase translation results are probably selected by tri-gram. We should remove improper functions from phrase patterns. Moreover, selecting the phrase pattern with semantic code will be required.

5.2 Open-Test

Experimental Method. Three-hundred compound or complex sentences are used for pattern matching. If sentence pattern matching and phrase pattern matching succeed, translation is performed. The translation accuracy is assessed with the evaluation criteria in section 5.1.

Pattern Matching Results. Sixty-one sentences are matched to sentence patterns. In these sentences, 14 sentences are matched to phrase patterns and 47 are not.

In the results of pattern matching, the pattern matching rate is about 5%. The reason for this seems to be the difference between input sentence styles and sentence pattern styles. Pattern matching is great influenced by the expression at the end of a sentence.

Phrase Pattern Matching. The main reason for failed phrase pattern matching is that the matched phrase string is longer than we expected. Fig. 10 shows this example.

Input Sentence	大阪までの間のどこかで駅弁を買って食べよう。 [Oosaka made no aida no dokoka de ekiben wo ka tte tabe you.]
Japanese Pattern	< <i>N1</i> は >! <i>VP2</i> (て で) <i>VP3</i> .you. [< <i>N1</i> wa>! <i>VP2</i> (te de) <i>VP3</i> .you.]
Matched Morph.	<i>VP2</i> = Oosaka made no aida no dokoka de ekiben wo ka <i>VP3</i> = tabe

Fig. 10. Examples of phrase pattern mismatch

In SPM-s, “大阪までの間のどこかで駅弁を買っ [Oosaka made no aida no dokokade de ekiben wo ka]” is matched as *VP*. It is longer than our phrase patterns. If there were the Japanese pattern “*ADV*P1の*VP*2て*V*3.you [*ADV*P1 no *VP*2 te *V*3.you]”, in the above sentence, “大阪までの間 [Oosaka made no aida]” would be matched as *ADV*P1, “どこかで駅弁を買っ [dokoka de ekiben wo katt]”, would be matched as *VP*2, and “食べ [tabe]” would be matched as *V*3. Then, phrase pattern matching would succeed.

Sentence Pattern Matching. Similarly, if we add sentence patterns, the sentence pattern matching rate is improved. However, it is expensive to add sentence patterns. It is found out that the sentence pattern matching rate with clause-level patterns is 78% (234 sentences are matched to the same 300 sentences). To increase the sentence pattern matching rate, the clause-level pattern-based translation should be implemented.

Human Evaluation. Table 4 shows results.

Table 4. Evaluation results

	Eval.1	Eval.2	Eval.3
#Sentence	1	9	4

The number in Eval. 2 is the largest, which is similar to the results of the closed-test. The main reason for the low translation accuracy is, similar to the closed-test, selection of the improper phrase translation.

6 Conclusion

We presented the phrase-level pattern-based translation system. Our method is based-on the Analogical Mapping Method. In the evaluation, the translation accuracy of phrase was still low, caused by improper phrase selection. Moreover, pattern matching rates of sentence and phrase are low.

For future work, we will resolve these problems of low translation accuracy. Also, we will implement the clause-level pattern-based translation to improve the low pattern matching rates.

References

1. Ikehara, S., Tokuhisa, M., Murakami, J.: Analogical Mapping Method and Semantic Categorization of Japanese Compound and Complex Sentence Patterns. In: Proceedings of the 10th Conference of the Pacific Association For Computational Linguistics, pp. 181–190 (2007)
2. Tokuhisa, M., Murakami, J., Ikehara, S.: Pattern Search by Structural Matching from Japanese Compound and Complex Sentence Pattern Dictionary. IPSJ SIG Technical Report, 2004-NL-176, pp. 9–16 (2006) (in Japanese)
3. Ikehara, S., Tokuhisa, M., Murakami, J., Saraki, M., Miyazaki, M., Ikeda, N.: Pattern Dictionary Development based on Non-Compositional Language Model for Japanese Compound and Complex Sentences. In: Matsumoto, Y., Sproat, R.W., Wong, K.-F., Zhang, M. (eds.) ICCPOL 2006. LNCS (LNAI), vol. 4285, pp. 509–519. Springer, Heidelberg (2006)
4. Tokuhisa, M., Endo, K., Kanazawa, Y., Murakami, J., Ikehara, S.: Evaluation of Pattern Generalization Effect under Development of Pattern Dictionary for Machine Translation. In: Pacific Association For Computational Linguistic, pp. 311–318 (2005)
5. Shapiro, S.C.: Generalized Augmented Transition Network Grammars For Generation From Semantic Networks. *Computational Linguistics Archive* 8(1), 12–25 (1982)
6. Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., Hayashi, Y.: *Goi-Taikei: A Japanese Lexicon*. Iwanami Shoten (1997) (in Japanese)

Parallel Texts Extraction from Multimodal Comparable Corpora

Haithem Affi, Loïc Barrault, and Holger Schwenk

Universit du Maine,
Avenue Olivier Messiaen F-72085 - LE MANS, France

Abstract. Statistical machine translation (SMT) systems depend on the availability of domain-specific bilingual parallel text. However parallel corpora are a limited resource and they are often not available for some domains or language pairs. We analyze the feasibility of extracting parallel sentences from multimodal comparable corpora. This work extends the use of comparable corpora by using audio sources instead of texts on the source side. The audio is transcribed by an automatic speech recognition system and translated with a baseline SMT system. We then use information retrieval in a large text corpus in the target language to extract parallel sentences. We have performed a series of experiments on data of the IWSLT'11 speech translation task that shows the feasibility of our approach.

Keywords: statistical machine translation, automatic speech recognition, multimodal comparable corpora, extraction of parallel sentences.

1 Introduction

The construction of a statistical machine translation (SMT) requires parallel corpus for training the translation model and monolingual data to build the target language model. A parallel corpus, also called bitext, consists in bilingual/multilingual texts aligned at the sentence level.

Unfortunately, parallel texts are a sparse resource for many language pairs with exception of English, French, Spanish, Arabic, Chinese and some European languages [6]. Furthermore, these corpora are mainly derived from parliamentary proceedings and news wire texts or produced by the United Nations. For the field of statistical machine translation, this can be problematic, because translation systems trained on data from a specific domain (*e.g.*, news) will perform poorly when applied to other domains, *e.g.* scientific articles.

One way to overcome this lack of data is to exploit comparable corpora which are much more easily available [9]. A comparable corpus is a collection of texts composed independently in the respective languages and combined on the basis of similarity of content. These are documents in one to many languages, that are comparable in content and form in various degrees and dimensions. Potential sources of comparable text corpora are multilingual news organizations such as Agence France Presse (AFP), Xinhua, Reuters, CNN, BBC, etc.. These texts

are widely available on the Web for many language pairs [13]. The degree of parallelism can vary considerably, from noisy parallel texts, to quasi parallel texts [3]. The ability to detect these parallel pairs of sentences enables the automatic creation of large parallel corpora.

However, for some languages, text comparable corpora may not cover all topics in some specific domains. What we need is to explore other sources like audio to generate parallel texts for each domain.

In this paper, we explore a method for generating parallel sentences from multimodal comparable corpus (audio and text). We would expect a useful technique to meet three criteria:

- Feasibility: the multimodal comparable corpora is useful to extract parallel text.
- Good quality: the quality of the parallel text generated from multimodal corpora should be comparable with bitext extracted from text comparable corpora.
- Effectiveness: since one of our motivations for exploiting comparable corpora is to adapt a SMT system for a specific domain, extracted bitext needs to be useful to improve SMT performance.

In the following sections, we will first describe the related work on parallel text extraction from comparable corpora for SMT. In section 3, we will describe our method. Section 4 describes our experiments and results.

2 Related Work

In the machine translation community, there is a long-standing belief that "there are no better data than more data". Following this idea, many works have been undertaken for mining large amounts of data in order to improve SMT systems. Thus, there is already an extensive literature related to the problem of comparable corpora, although from a different perspective than the one taken in this paper.

Typically, comparable corpora don't have any information regarding document pair similarity. Generally, there exist many documents in one language which don't have any corresponding document in the other language. Also, when the corresponding information among the documents is available, the documents in question are not literal translations of each other. Thus, extracting parallel data from such corpora requires special algorithms designed for such corpora.

An adaptive approach, proposed by [19], aims at mining parallel sentences from a bilingual comparable news collection collected from the web. A maximum likelihood criterion was used by combining sentence length models and lexicon-based models. The translation lexicon was iteratively updated using the mined parallel data to get better vocabulary coverage and translation probability estimation. In [18], an alignment method at different levels (title, word and character) based on dynamic programming is presented. The goal is to identify the one-to-one title pairs in an English/Chinese corpus collected from the web,

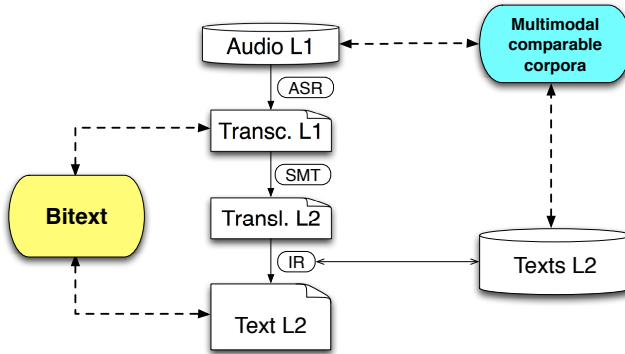


Fig. 1. Extracting parallel texts from multimodal comparable corpora

They applied longest common sub-sequence (LCS) to find the most reliable Chinese translation of an English word. [13] propose a web-mining based system called STRAND and show that their approach is able to find large numbers of similar document pairs.

[17] uses cross-language information retrieval techniques and dynamic programming to extract sentences from an English/Japanese comparable corpus. They identify similar article pairs, and then, considering them as parallel texts, they align their sentences using a sentence pair similarity score and use DP to find the least-cost alignment over the document pair.

[9] uses a bilingual lexicon to translate some of the words of the source sentence. These translations are then used to query the database to find matching translations using information retrieval (IR) techniques. [1] bypass the need of the bilingual dictionary by using proper SMT translations. They also use simple measures like word error rate (WER) or translation edit rate (TER) in place of a maximum entropy classifier.

In another way, [12] demonstrated that statistical translation models can be trained in a fully automatic manner from audio recordings of human interpretation scenarios.

In this paper, we are interested in generating a parallel text from a comparable corpora composed by an audio part in one language and a text part in other language. To the best of our knowledge, no systematic empirical research exists addressing the use of comparable audio corpora to extract bitexts.

3 Extracting Parallel Texts from Multimodal Comparable Corpora

Our main experimental framework is designed to address the situation when we translate data from a domain different than the training data. In such a condition, the translation quality is generally rather poor.

In this work we seek to improve SMT systems in domains that suffer from resource deficiency by automatically extracting bitexts from an comparable corpora with include audio. We propose an extension of the methods described in [1]. The basic system architecture is described in Figure 1. We can distinguish three steps: automatic speech recognition (ASR), statistical machine translation (SMT) and information retrieval (IR). The ASR system accepts audio data in language L1 and generates an automatic transcription. This transcription is then translated by a baseline SMT system into language L2. Then, we use these translations as queries for an IR system to retrieve most similar sentences in the text part of our multimodal comparable corpus. The transcribed text in L1 and the IR result in L2 form the final bitext. We hope that the errors made by the ASR and SMT systems will not impact too severely the quality of the IR queries, and that the extracted bitext will improve an SMT system.

3.1 Task Description

This framework raises several issues. Each step in the system can introduce a certain number of errors. It is important to highlight the feasibility of the approach and the impact of each module on the generated data. Thus, we conducted three different types of experiments, described in Figure 2. In the first experiment (*Exp 1*) we use the reference translations as queries for the IR system. This is

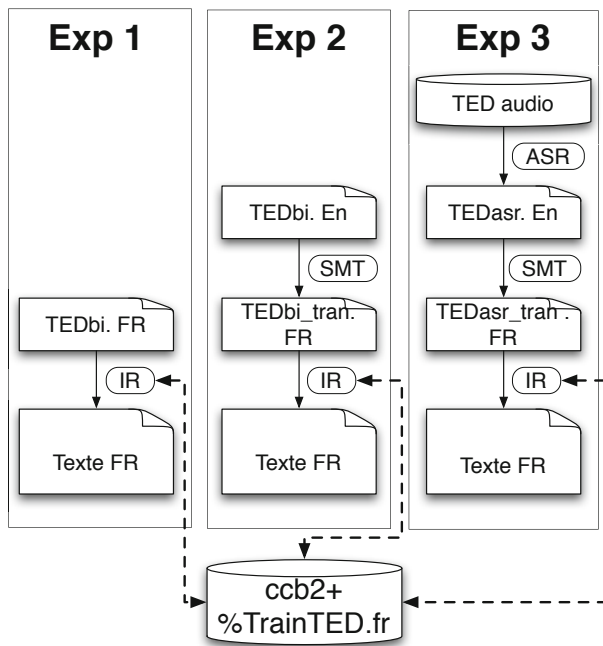


Fig. 2. Different experiments to analyze the impact of the errors of each module

the most favorable condition, it simulates the case where the ASR and the SMT systems do not commit any error. In the second experiment (*Exp 2*) we use the reference transcription as input to the SMT system. In this case, the errors come only from the SMT system since no ASR is involved. Finally, the third experiment (*Exp 3*) represents the complete proposed framework, described in section 3. It corresponds to a real scenario.

Another issue is the importance of the degree of similarity between the two parts of the comparable corpora. In a real life comparable corpus, we can only expect to find matching sentences for a fraction of the sentences. Therefore, we artificially created four comparable corpora with different degrees of similarity. The source part of our comparable corpus is always the TED corpus (see next section). The target language part of the comparable corpus consists of a large generic corpus plus 25%, 50%, 75% and 100% respectively of the reference translations of the TED corpus.

For each candidate sentence pair, we need to decide whether the two sentences in the pair are mutual translations. Thus, we classify the IR result with TER [15] calculated between the query, i.e. the automatic translation, and the sentence selected by IR.

In all cases, an evaluation of the approach is necessary. Thus, the final parallel data extracted are re-injected into the baseline system. The various SMT systems are evaluated using the BLEU score [11]. This is the most commonly used metric in the domain of automatic machine translation, but the choice of the best metric is actually still an open research issue.

4 Experimental Setup

4.1 Data Description

Our comparable corpus consist of two monolingual corpora, one spoken in English and one written in French. In our experiments we use all available data from IWSLT'11 evaluation campaign. The goal of the so-called TED task¹ is to translate public lectures from English into French. The TED corpus totals about 118 hours of speech. A detailed description can be found in [14].

For MT training, we considered the following corpora among those available: the latest versions of News-Commentary (nc7) and Europarl (eparl7) corpus, the TED corpus provided by IWSLT'11 (*TEDbi*) and a subset of the French-English 10⁹ Gigaword corpus (ccb2). The Gigaword corpus was filtered with the same techniques described in [14]. We name it ccb2_px70. We transcribed all the TED audio data with the ASR system described in section 4.2 and name it *TEDasr*. Table 1 summarizes the characteristics of those different corpora. Each corpus is labeled whether it is in- or out-of domain with respect to our task.

The development corpus (dev) consists of 19 talks and represents a total of 4 hours and 13 minutes of speech. We use the same test data as provided by IWSLT'11 for the speech translation task. *dev.outASR* and *test.outASR* are the

¹ <http://www.ted.com/>

Table 1. MT training (left) and development data (right)

bitexts	# words	in-domain ?	Dev	# words
nc7	3.7M	no	dev.outASR	36k
eparl7	56.4M	no	dev.refSMT	38k
ccb2_px70	1.3M	no	Test	# words
TEDbi	1.9M	yes	tst.outASR	8.7k
TEDasr	1.8M	yes	tst.refSMT	9.1 k

automatic transcriptions of the development and test corpus respectively. The reference translations are named *dev.refSMT* and *tst.refSMT*. Table 1 summarizes the characteristics of the different corpora used in our experiments.

4.2 ASR System Description

Our ASR system is a five-pass system based on the open-source CMU Sphinx toolkit (version 3 and 4), similar to the LIUM’08 French ASR system described in [2]. The acoustic models were trained in the same manner, except that a multi-layer perceptron (MLP) is added using the Bottle-Neck feature extraction as described in [5]. Table 2 shows performances of ASR system on the dev and test corpora.

Table 2. Performances of the ASR system on dev and test data (% WER)

Corpus	% WER
dev.outASR	19.2%
test.outASR	17.4%

4.3 SMT System Description

Our system is a phrase-based system [8] which uses fourteen features functions, namely phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, a word and a phrase penalty and a target language model. It is based on the Moses SMT toolkit [7] and is constructed as follows. First, word alignments in both directions are calculated. We used the multi-threaded version of the GIZA++ tool [4]. Phrases and lexical reordering are extracted using the default settings of the Moses toolkit. The parameters of our system were tuned on *dev.outASR*, using the MERT tool. The language model was trained with the SRI LM toolkit [16], on all the French data distributed in IWSLT 2011 evaluation campaign without the TED data. The baseline system is trained with eparl7 and nc7 bitexts.

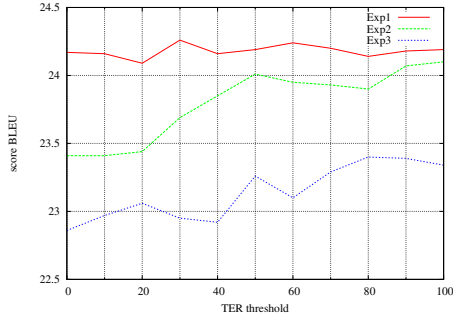


Fig. 3. BLEU score on dev using SMT systems adapted with bitexts extracted from *ccb2* + 100% *TEDbi* index corpus

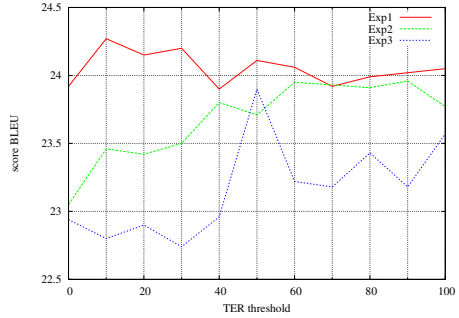


Fig. 4. BLEU score on dev using SMT systems adapted with bitexts extracted from *ccb2* + 75% *TEDbi* index corpus

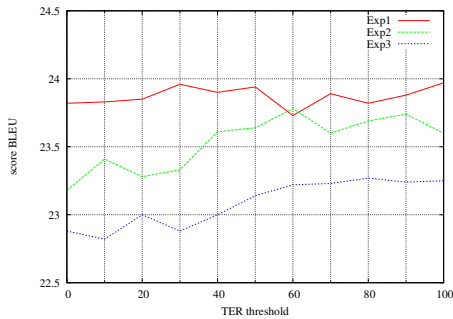


Fig. 5. BLEU score on dev using SMT systems adapted with bitexts extracted from *ccb2* + 50% *TEDbi* index corpus

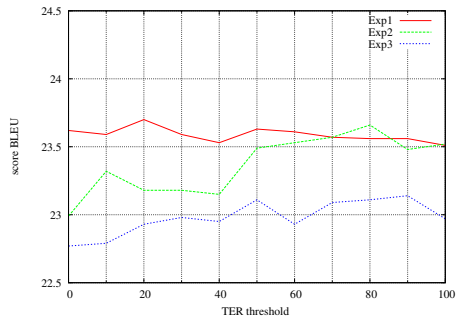


Fig. 6. BLEU score on dev using SMT systems adapted with bitexts extracted from *ccb2* + 25% *TEDbi* index corpus

4.4 IR System

We used the Lemur IR toolkit [10] for the sentence extraction procedure. We first index all French text data into a database using *Indri Index*. This feature enabled us to index our text documents in such a way that using the specialized *Indri Query Language* we can use the translated sentences as queries to run TF-IDF retrieval in the database. By these means we can retrieve the best matching sentences from the French side of the comparable corpus. The index data consist of the French part of *ccb2*_px70 and different percentage of the French side of *TEDbi* as described in section 3.1.

4.5 Experimental Results

As mentioned in section 3, the TER score is used as a metric for filtering the result of IR. We keep only the sentences which have a TER score below a certain

threshold determined empirically. Thus, we filtered the selected sentences in each condition with different TER thresholds ranging from 0 to 100. The extracted bitexts were added to our generic training data in order to adapt the baseline system. The Figures 3, 4, 5 and 6 present the BLEU score obtained for these different experimental conditions.

In *Exp2*, we use automatic translations for the IR queries. One can hope that IR itself is not too much affected by the translation errors, but this will be of course the fact for the filtering based on the TER score. [1] propose to vary the TER threshold between 0 and 100 and to keep the threshold value that maximizes the BLEU score once the corresponding extracted bitexts were injected into the generic system. We did not observe such a clear maximum in our experiments and the BLEU score increases almost continuously. Nevertheless, in order to limit the impact of noisy sentences, we decided to only keep the sentences with a TER score below the threshold of 80. One can observe that the BLEU score of the adapted system matches the one of *Exp1* in most of the cases. Therefore, we conclude that the errors induced by the SMT system have no major impact on the performance of the parallel sentence extraction algorithm. These findings are in line with those of [1].

These results show that the choice of the appropriate TER threshold depends on the type of data. Our baseline SMT system trained with generic bitext only achieves a BLEU score of 22.93. In *Exp1*, we use the reference translations as query and IR should in theory find all the sentences in the large corpus with a TER of zero. It can happen that our generic ccb2 corpus also contains some similar sentences which are “accidentally” retrieved. The four figures show that IR does indeed work as expected: the observed improvement in the BLEU score does not depend on the TER threshold (with the exception of some noise) since all the sentences have a TER of zero. The achieved improvement depends of course on the amount of TED bitexts that are injected in our comparable corpus: the BLEU increases from 22.93 to 24.14 when 100% is injected while we only achieve a BLEU score of 23.62 when 20% is injected. These results give us the upper bound that we could expect to get when extracting parallel sentences from our multimodal comparable corpus.

Finally, in *Exp3*, we use automatic speech recognition on the source side of the comparable corpus. Our ASR system has a WER of about 18%. These errors on the source side can obviously lead to wrong translations and have a negative impact on the IR process. It is important to note that these automatic transcriptions represent the source side of our extracted parallel corpus. By these means, eventual transcription errors should less affect the translation system since it is unlikely that wrong source phrases will be used to translate other texts. We observed in our experiments that the extracted sentences do improve the SMT system. The performance is actually only 0.5 BLEU points below those obtained in *Exp1* or *Exp2*.

Table 3 lists the adaptation results of the baseline system in different conditions. It shows that starting with a BLEU score of 23.96% on the test set for the baseline system, adaptation with automatically extracted in-domain bitext

Table 3. BLEU scores on dev and test after adaptation of a baseline system with bitexts extracted in conditions *Exp1*, *Exp2* and *Exp3* (100% TEDbi)

Experiment	Dev	Test
Baseline	22.93	23.96
Exp1	24.14	25.14
Exp2	23.90	25.15
Exp3	23.40	24.69

resulted in an improvement in all conditions between 1.18 in *Exp1* and 0.73 BLEU points in *Exp3*.

Table 4 provides an analysis of the performance in function of the degree of parallelism of the comparable corpus. Remember that the whole corpus amounts to about 1.8M words. We were able to extract automatically about 400k words of new bitexts, i.e. a little more than 20%. If less data is injected, the amount of extracted data decreases linearly.

Table 4. BLEU scores for different degrees of parallelism of the comparable corpus

Experiments	Dev	Test	# injected words
Baseline	22.93	23.96	-
25% TEDbi	23.11	24.40	~110k
50% TEDbi	23.27	24.58	~215k
75% TEDbi	23.43	24.42	~293k
100% TEDbi	23.40	24.69	~393k

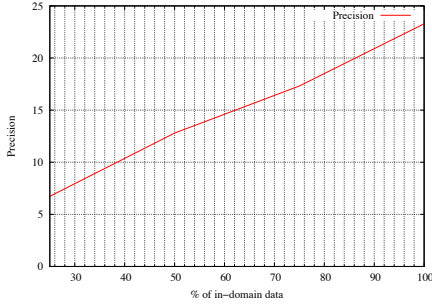
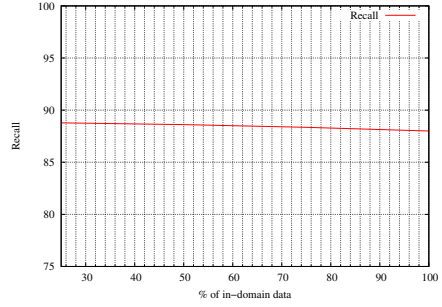
We have also measured the performance of the extraction process by computing the precision and recall. Precision is computed as the ratio of sentence pairs correctly identified as parallel over the total number of sentence pairs extracted (for a given TER threshold). Recall is computed as the ratio of parallel sentence pairs extracted by the extraction system to the total number of sentences i.e., in-domain injected (*TEDbi*) and out-of-domain (*ccb2*). Both are expressed as percentages:

$$Precision = \frac{100 * nb \text{ parallel sentences retrieved}}{total \text{ nb sentences extracted}} \quad (1)$$

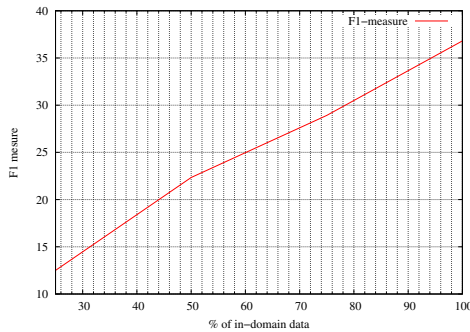
$$Recall = \frac{100 * nb \text{ parallel sentences extracted}}{total \text{ nb sentences bitext}} \quad (2)$$

The combination of the two measures with an equal weight gives the F-measure, presented by the following expression:

$$F = 2 \frac{Recall \cdot Precision}{Recall + Precision} \quad (3)$$

**Fig. 7.** Precision of the system extraction**Fig. 8.** Recall of the system extraction

As we can see in Figure 8, the value of the Recall is stable because we extract the same number of sentences in all of our experiments. We can clearly see in Figure 9 that the performance in terms of F-measure of our system extraction depends on the degree of parallelism of the comparable corpus. This curve validates the previous results in terms of the BLEU score.

**Fig. 9.** F-measure of the system extraction

We argue that this is an encouraging result since we automatically aligned source audio in one language with texts in another language, without the need of human intervention to transcribe and translate the data. The TED corpus contains only 118 hours of speech. There are many domains for which much larger amounts of untranscribed audio in one language and related texts in another language are available, for instance news.

5 Conclusion

Domain specific parallel data is crucial to train well performing SMT systems, but it is often not easily and freely available. During the last years, there are

several works that propose to exploit comparable corpora for this purpose and many algorithms were proposed to extract bitexts from a comparable corpus.

In this paper, we have proposed to extend this concept to multimodal comparable corpora, i.e. the source side is available as audio and the target side as text. This is achieved by combining a large vocabulary speech recognition system, a statistical machine translation system and information retrieval. We validate the feasibility of our approach by a set of experiments to analyze the impact of the errors committed by each module. We were able to adapt a generic SMT system to the task of lecture translation by extracting parallel data from a multimodal comparable corpus composed of 118 hours of untranscribed speeches in the source language and 1.8M words of translations injected into a large generic corpus. This led to an improvement of 0.7 in the BLEU score.

Acknowledgments. This work has been partially funded by the French Government under the project DEPART.

References

1. Abdul-Rauf, S., Schwenk, H.: Parallel sentence generation from comparable corpora for improved SMT. *Machine Translation* (2011)
2. Deléglise, P., Estève, Y., Meignier, S., Merlin, T.: Improvements to the LIUM french ASR system based on CMU Sphinx: what helps to significantly reduce the word error rate? In: *Interspeech 2009*, September 6-10 (2009)
3. Fung, P., Cheung, P.: Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In: *Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004* (2004)
4. Gao, Q., Vogel, S.: Parallel implementations of word alignment tool. In: *Software Engineering, Testing, and Quality Assurance for Natural Language Processing, SETQA-NLP 2008*, pp. 49–57 (2008)
5. Grézl, F., Fousek, P.: Optimizing bottle-neck features for LVCSR. In: *2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4729–4732. *IEEE Signal Processing Society* (2008)
6. Hewavitharana, S., Vogel, S.: Extracting parallel phrases from comparable data. In: *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web, BUCC 2011*, pp. 61–68 (2011)
7. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: open source toolkit for statistical machine translation. In: *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 177–180 (2007)
8. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, vol. 1, pp. 48–54 (2003)
9. Munteanu, D.S., Marcu, D.: Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *Computational Linguistics* 31(4), 477–504 (2005)
10. Ogilvie, P., Callan, J.: Experiments using the lemur toolkit. In: *Proceeding of the Tenth Text Retrieval Conference, TREC-10* (2001)

11. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318 (2002)
12. Paulik, M., Waibel, A.: Automatic translation from parallel speech: Simultaneous interpretation as MT training data. In: ASRU, Merano, Italy (December 2009)
13. Resnik, P., Smith, N.A.: The web as a parallel corpus. *Comput. Linguist.* 29, 349–380 (2003)
14. Rousseau, A., Bougares, F., Deléglise, P., Schwenk, H., Estève, Y.: LIUM’s systems for the IWSLT 2011 speech translation tasks. In: International Workshop on Spoken Language Translation 2011 (2011)
15. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of Association for Machine Translation in the Americas, pp. 223–231 (2006)
16. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: International Conference on Spoken Language Processing, pp. 257–286 (November 2002)
17. Utiyama, M., Isahara, H.: Reliable measures for aligning japanese-english news articles and sentences. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 72–79 (2003)
18. Yang, C.C., Li, K.W.: Automatic construction of english/chinese parallel corpora. *J. Am. Soc. Inf. Sci. Technol.* 54, 730–742 (2003)
19. Zhao, B., Vogel, S.: Adaptive parallel sentences mining from web bilingual news collection. In: Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM 2002, 745 pages. IEEE Computer Society, Washington, DC (2002)

A Reliable Communication System to Maximize the Communication Quality

Gan Jin and Natallia Khatseyeva

Centre Tesnière, Université de Franche-Comté, Besançon, France
gan.jin@edu.univ-fcomte.fr,
xateeva@gmail.com

Abstract. Chinese and Russian differ a lot from the European languages, e.g. French, regarding their morphology, lexicon, syntax and semantics. This complexity causes a lot of problems for Machine Translation (MT) systems. This paper describes our MT system, operating from French to Chinese and Russian, and reveals through different examples our methodology. The latter is designed to deal with this complexity, in order to increase the accuracy of the human communication.

Keywords: Machine Translation, linguistics, Chinese, Russian, ambiguity.

1 Introduction

In the actual world, where trade and communication are carried out at the international level even within one and the same company or institution, the presence of the foreign language exchanges is evident. Regarding the security domain, a correct communication of warning and safety messages is even mandatory, since their content must be transmitted without any fault or error. For this purposes, a multilingual security messages Machine Translation (MT) system, as part of communication software, is of great interest. However, it must be based on a relevant theory to provide unambiguous and perfectly correct translations, as any mistake in the security area may be fatal.

2 Machine Translation Methodology

There are two main MT methodologies: the rule-based and the corpus-based (presented by statistical-based systems). Direct systems are known as MT systems of the first generation, while systems of the second generation, as opposed to the corpus-based, are designed on specified rules of morphologic, syntactic, lexical and semantic analysis and are generally known as rule-based systems.

As our methodology presents a linguistic-oriented theory, we will develop its details below. There are 3 main linguistic approaches to MT:

- direct;
- interlingual or pivot;
- transfer.

The first is called 'direct translation approach' or 'binary translation'. MT systems of this type are designed in detail for a concrete pair of languages. The translation from the source to the target language is direct, with little syntactic and semantic analysis, if necessary. The source language analysis is made depending on the chosen target language.

Another type of MT approaches is represented by the 'interlingual systems', based on the hypothesis that different source texts can be converted into one intermediate source language with syntactic and semantic representations common to several treated languages. Target languages will be then generated from these representations. The translation is being made in 2 steps: from the source language to the interlingua, and from the interlingua to the target languages. Analysis procedures are called specific to the source language and are not oriented to any precise target language. Similarly, the target language synthesis (generating) rules are not created to respond any source language particularities. This methodology is mainly useful for multilingual domains, because it does not require further source language modifications, when a new target language is being added, and the previously built generating programs can be used. But despite the complexity of these pivot systems, we have only partly used this methodology. We will develop this idea later.

The third MT approach, representing the transfer systems, is less ambitious. Comparing to the pivot approach it operates in three steps instead of two, using abstract representations of both source and target languages. During the first step the source text is converted into abstract representations, then during the second step these representations are converted to the corresponding target language ones, and then finally during the third step the generation is being performed.

Speaking about our system, we partly imitate both the interlingual/ pivot and the transfer methodologies, more precisely EBMT (Exemple based machine translation) [1]. To perform our MT we obtain a logical source language representation, which, instead of leading us directly to our target languages, undergoes a series of minimal transformations. The transformations are being made in the same logical language that erases ambiguities between the source and the target language and permits to produce exact equivalents between these languages. In the same time, the generation phase is also facilitated, due to the fact that the pivot and the other language difference representations are made in the same logical language. We should emphasize that our pivot language reflects from the start the particularities of our source and different target languages (which is possible due to our prior norms analysis) [2].

Regarding the actual state of MT knowledge, the problem consists in everyone's desire to label methodologies, which leads to a phase where people do not know their exact content anymore (about this and for more EBMT information see [3]).

3 Our Methodology

Since our chosen methodology has already given birth to the 'Traduction Automatique Centre Tesnière' (TACT) MT system, we use its architecture. The principles of this architecture are based on the SyGULAC theory, which is rooted in micro-systemic analysis and discrete mathematics. The methodology that results from this theory leads to three applications: controlled languages, MT and data sense-mining [5].

Micro-systemic linguistic analysis, which originated with the analysis of linguistic norms, [6] is based on the postulate that a language can be segmented into individual systems based on the observation that such systems influence each other.

Micro-systemic linguistic analysis proposes that to be processed languages have to be decomposed into systems which can be analysed by a human being and by machine, because they are small enough but also complete so as to be able to work together as a unified system. Besides this, the systems, which are so delimited, can interact with other similar systems, and this interaction is a property of language. Nothing is independent; lexis, morphology, syntax are linked [7].

Micro-systemic linguistic analysis consists in analysing a linguistic system in component systems as follows:

- Sc: a system, which is recognisably canonical.
- Sv: another system representing the variants.
- Ss: a 'super' system, which puts the two systems Sc and Sv in relation with each other.

To establish system Ss for some application, the linguist establishes two categorisations and then puts these into relation, one with the other. The categorisations are:

- i. a 'non-contextual' (nc) categorisation of the canonical forms in relation with the variant forms in isolation, the context being limited to just the canonical and variant forms themselves.
- ii. an 'in-context' (ic) categorisation of the canonical forms in relation with the variant forms in terms of the linguistic contexts of the variant forms [7].

The limit 'canonical' case where there are no divergences corresponds to the target language being identical to the source language. In our case this language is the French controlled language, which has also been controlled for machine translation to the specific target controlled languages. Thus our controlled languages mirror each other [8]. The architecture of our MT system is thus based on this canonical limit case with the divergences between our target controlled languages and our controlled French being 'variant' cases, which are organized in such a manner as to effect the translations during the translation process [7].

4 Controlled Language

Our chosen methodology is characterised by pre-editing absence, as both the controlled source and target languages initially remove the ambiguities and anything that can affect the clarity of information.

Therefore, a controlled language (CL) is a simplified version of a natural standard language that possesses restrictive rules on its syntax and lexicon. The main purpose of controlling the language is to increase text comprehension and readability, to limit possible errors of interpretation and thus to facilitate MT. LC restrictive rules can sometimes shorten the treated texts, but it is still a structurally normal and understandable language. The most important part of it is that each control rule must be justified. Once all these rules are applied to our text data, we obtain a list of formalised macrostructures that can cover the sentences of our domain corpora.

5 Theoretical Model

As already mentioned, the biggest advantage of our system is that both the source and the target languages are controlled, which helps to avoid the pre-editing as well. In reality, the control is provided during message entry by means of the 'SL to CPSL' User Interface (Source Language to Controlled Pivot Source Language). That means that our MT system is a novel hybrid (pivot + transfer) rule-based machine translation architecture in which the pivot language (PL) is French controlled also for translation, and where the Transfer System is directed by the various source-target language divergences.

Another great advantage is that our system architecture is made in the way, that it remains fully extendable and requires minimal changes when adding a new target language. At the same time it obviously needs a new complete linguistic module, containing a dictionary and a complete micro-systemic representation of this new target language. So, once our system is prepared for the new target language adding, we can proceed to the formalisation of our controlled Russian/ Chinese to design its linguistic model.

Table 1, Table 2 and Table 3 following illustrate how transfer Ss operates: first between controlled source French and controlled pivot French, and then between controlled pivot French and controlled target Russian/ controlled target Chinese. They also show how divergences are incorporated to our system and how connections are made at the 'super'-level.

Table 1. Ss_frC_frC: Controlled French → Controlled French (Identity – no divergences, constructed from Ss-frC)

Table					
frC					
frC_groupesVerbaux_frC					
frC_args_frC					
frC_groupes					
frC_dictionnaireLexical_frC					
frC_catégories_frC					
frC_catégories					
frC_groupesVerbaux_frC					
verbe_frC		verbe_frC		groupe_frC	exemple
frC_args_frC					
arg_frC			arg_frC	exemple	
frC_groupes					
groupe_frC			structure_frC	exemple	
frC_dictionnaireLexical_frC					
lexique_frC		catégorie_frC_frC		lexique_frC	catégorie_frC
frC_catégories_frC					
catégorie_frC_frC			exemple		
frC_catégories					
catégorie_frC			exemple		

Table 2. Ss_frC_ruC: Controlled French → Controlled Russian

Table					
ruC					
ruC_groupesVerbaux_frC					
...					
ruC_groupesVerbaux_frC					
verbe_frC		verbe_ruC	groupe_ruC	exemple	corpus
ruC_args_frC					
arg_frC			arg_ruC	exemple	corpus
ruC_groupes					
groupe_ruC			structure_ruC	exemple	corpus
ruC_dictionnaireLexical_frC					
lexique_frC		catégorie_frC_ruC	lexique_ruC	noLex_ruC	ge-Lex_ruC
ruC_catégories_frC					
catégorie_frC_ruC			exemple		corpus
ruC_catégories					
catégorie_ruC			exemple		corpus

Table 3. Ss_frC_ruC: Controlled French → Controlled Chinese

Table					
chC					
chC_groupesVerbaux_frC					
...					
chC_groupesVerbaux_frC					
verbe_frC		verbe_chC	groupe_chC	exemple	corpus
chC_args_frC					
arg_frC		arg_chC		exemple	corpus
chC_groupes					
groupe_chC		structure_chC		exemple	corpus
ruC_dictionaireLexical_frC					
lexique_frC		catégorie_frC_chC		lexique_chC	
chC_catégories_frC					
catégorie_frC_chC		exemple		corpus	
chC_catégories					
catégorie_chC		exemple		corpus	

To present the application of our theory more transparently, we will consider a sentence with a fairly large number of analysis and generation problems, which should be resolved to enable the MT to our target languages (Chinese and Russian).

Mettre chaque manette de poussée en position IDLE immédiatement.

This sentence will be represented in our formal logical language in the following form:

$$\text{opt}(\text{neg1}) + \text{opt}(\text{neg2}) + \text{vinf}(\text{'mettre'}) + \text{arg1} + \text{prep}_v(\text{'en'}) + \text{arg2} + \text{opt}(\text{prep_comp}) + \text{opt}(\text{comp, Na}) + \text{opt}(\text{prep_comp}) + \text{opt}(\text{comp, Na})$$

During this step we obtain the segmentation as follows:

Mettre/ chaque manette de poussée/ en /position IDLE/ immédiatement.

The segmentation of our element 'Mettre chaque manette de poussée en position IDLE' is being made within our dictionary of verb structures. At this step the verb structure is the following:

$$\text{vinf('Mettre')} + \text{arg1}(qqc/qqn) + \text{prep}_v(\text{'en'}) + \text{arg2}(\text{position})$$

Actually, while the user was entering this sentence, the program choose this structure automatically among the following verb structures:

4 : $\text{opt}(\text{neg1}) + \text{opt}(\text{neg2}) + \text{vinf}(\text{'Mettre'}) + \text{arg1}(qqc)$

7 : $\text{opt}(\text{neg1}) + \text{opt}(\text{neg2}) + \text{vinf}(\text{'Mettre'}) + \text{arg1}(qqc/qqn) + \text{prep}(\text{'en'}) + \text{arg2}(\text{position})$

5 : $\text{opt}(\text{neg1}) + \text{opt}(\text{neg2}) + \text{vinf}(\text{'Mettre'}) + \text{arg1}(qqn) + \text{part}_v(\text{'sur respirateur'})$

4c : $\text{arg0}(qqn) + \text{opt}(\text{neg1}) + \text{vconj}(\text{'Mettre'}) + \text{opt}(\text{neg2}) + \text{arg1}(qqc)$

7c : $\text{arg0}(qqn) + \text{opt}(\text{neg1}) + \text{vconj}(\text{'Mettre'}) + \text{opt}(\text{neg2}) + \text{arg1}(qqc) + \text{prep}(\text{'sur'}) + \text{arg2}(qqc)$

At the user level structures 4 and 7 correspond to one and the same structure:

$$\text{opt}(\text{neg1}) + \text{opt}(\text{neg2}) + \text{vinf}(\text{'mettre'}) + \text{arg1}(qqc/qqn) + \text{opt}(\text{prep}(\text{'en'}) + \text{arg2}(\text{position}))$$

Once the user has validated the sentence, the system determines whether it corresponds to structure 4 or structure 7, depending the filled in elements (and it keeps the longest string).

There are also source or target language sentences, which do not reveal all the arguments explicitly, so it is necessary for the system to keep the places of these arguments, to be able to make the transfer from one language to another, even though they are blank. E.g. correspondences between our French and target verb structures in Chinese:

$$\text{indicateur_chC}(\text{'在'} \text{ 摁}) + \text{arg1} + v + \text{arg2}$$

and in Russian:

$$v + \text{arg1_acc} + \text{prep}_v(\text{'в'}) + \text{arg2_acc}$$

The French structure in preparation for Chinese and the Chinese structure:

$$n + \text{adj} \rightarrow \text{adj} + n$$

'position IDLE' = 'IDLE位置'

The French structure in preparation for Russian and the Russian structure:

$$n + \text{adj} \rightarrow n + \text{adj}$$

'position IDLE= 'положение IDLE'

For the dictionary explanation we will give one complex example, which reveals that the arguments representation is not always the same in the source and all the target languages. Some of its components may not exist, but for our system tracking purposes we also keep the emplacement of the omitted elements (as with the arguments).

Here you can see a part of our dictionary that can help you to understand its format and how the omitted elements placement is kept. The representation in both Chinese and Russian will also reveal you the differences between several target languages.

Table 4. Chinese dictionary

French lexical entry	French category	Chinese lexical entry	Chinese category
à	prep_comp	∅	∅
la	ad	∅	∅
maison	n	家	n
immédiatement	adv	马上	adv

Table 5. Russian dictionary

French lexical entry	French category	Russian lexical entry	Russian category
à	prep_comp	∅	∅
la	ad	∅	∅
maison	n	дома	adv
immédiatement	adv	немедленно	adv

Now we will consider another significant example, taken from the corpus, that is being used within 'Airbus' company and that has been processed for them by the research group of Centre Tesnière, University of Franche-Comté (France).

Appuyer sur le bouton clignotant orange maintenant.

The French verb structure for this sentence is the following: *vinf* + *prep_v* + *arg2* + *comp*.

And its corresponding Chinese structure representation is: *comp* + *v* + *arg2*.

Once our verb structure is defined, we can make the arguments representation and correspondence from French: 'bouton *clignotant orange*'=*n* + *adj1* + *adj2* to

Chinese: ‘黄色闪光的按钮’= adj2 + adj1 + n, to finally obtain the whole translation as follows: 现在按黄色闪光的按钮.

When translating this sentence into Russian, we meet more significant problems and have to deal not only with the Russian typical case according system, but also with some arguments placement modifications.

First of all, Chinese and Russian do both assign a new placement to the complement that should be precede the verb, so we obtain a mostly similar Russian verb structure:comp + vinf +prep_v + arg2.

But secondly, when drawing correspondences between argument representations in French and Russian we obtain the following structure: adj2_acc + adj1_acc +n_acc.

Where the adjectives are not only placed before the noun, but their order is also interchanged between them. So finally we obtain the Russian generated translation: Немедленно нажать на моргающую оранжевую кнопку.

6 Our system Prototype and Tests

As we have already explained, our controlled structure that should be found in the pivot language comes from the user interface. In case we process the sentence ‘Ne pas brancher trop d’appareils électriques sur la même prise à la maison’, its formalised structure will be the following:Nég brancher qqc sur qqc à lieu.

In reality, the preposition ‘sur’ is displayed automatically, because it is compulsory in the chosen verb structure, as ‘brancher’ is a verb, corresponding to French frC_7 group, and its structure is the following:opt(neg1) + opt(neg2) + vinf(‘brancher’) + arg1(qqc) + prep(‘sur’) + arg2(qqc).

This first verb structure representation figures in the pivot language, as it has been previously controlled according to French and then converted into French logical structures. It helps our source structures to be more similar to our target language representations for further logical language processing, in order to obtain in the end our target logical language structures.

Then each component is searched in French dictionaries to obtain its target language equivalents. For that purpose lexical entries and their grammatical categories are tagged depending each target language. What about the morphology, the generation can be more complex. Sometimes we may need agreement and dependency algorithms, which will generate affixes, case agreement morphemes or any other target language peculiarities automatically. Our grammars and rewriting rules specify different level constraints to end with a final target language translation.

To make ourselves sure of our system efficiency, we performed some rapid tests on our last example, using different MT systems:

Ne pas brancher trop d’appareils électriques sur la même prise à la maison.

The results for Chinese are the following:

WorldLingo

不连接许多electricals器具在同样捉住在房子

Google

不要插入太多电器在国内同插座

TACT (our system)
不要在家把太多的电器插在同一个插销上

And for Russian:

WorldLingo
Не подключить too many прибора electricals на этих же уловите на доме.

Google
Не подключайте слишком много приборов в тот же сокет дома.

TACT (our system)
Не подключать слишком много электрических приборов в одну и ту же розетку дома.

When looking at the obtained Russian translation results, with 'WorldLingo' system we obtain a catastrophically wrong and not agreed sentence, where only the verb is translated correctly. And what about 'Google' translation results, they are not complete, with an omitted adjective 'электрических', and with an incorrect noun translation 'сокет' instead of 'розетку', coming from a definitely different speciality domain.

In the end, as we can notice, our system is the most relevant in the domain of security MT.

7 Conclusions

As we have already mentioned, the necessity of warning and safety message MT systems is evident, and to be useful and correct it just has to be based on a relevant theory.

The major advantage of our system is that technically it does not need a lot of volumes to be installed, as it is a rule-based and not statistical system, so it avoids the storage of bilingual translation corpuses.

And the novelty and advantage of our methodology is due to the fact that:

- it provides accurate results without translation revision;
- it is a rule-based linguistic-oriented method with a mixed approach combining: pivot to dependency grammar using source language predicates/ arguments and controlled source and target languages; pivot to dependency grammar using predicates/ arguments of each target language, that permits to reveal and treat the differences between the languages; and use of transfer method at the argument level.

These two pivots emphasize the similarities between our studied languages, a thing noticed due to the performed control and to our language norms research and implementation during the MT system elaboration (something that we doubt has existed before) [4].

References

1. Cardey, S., Kiattibutra, R., Beddar, M., Devitre, D., Gentilhomme, S., Greenfield, P., Jin, G., Mikati, Z., Renahy, J., Sekunda, G.: Modèle pour une Traduction Automatique fi-dèle, le système TACT multilingue, Projet LiSE ANR (Linguistique et Sécurité). In: Actes du WISG 2009, Workshop Interdisciplinaire sur la Sécurité Globale, CD-ROM, Janvier 28-29, 10 pages. Université de Technologie de Troyes (2009)
2. Cardey, S., et al.: Les langues contrôlées, fondements, besoins et applications. In: Actes du WISG 2008, Workshop Interdisciplinaire sur la Sécurité Globale (CD ROM), Janvier 29-30, 10 pages. Université de Technologie de Troyes (2008)
3. Hutchins, J.: Towards a definition of example-based machine translation. In: MT Summit X, Phuket, Thailand: Proceedings of Workshop on Example-Based Machine Translation, September 16, pp. 63–70 (2005)
4. Cardey, S.: La théorie systémique et ses calculs SyGULAC appliqués à la syntaxe du français. In: Acte du Colloque "Problèmes Contemporains de l'histoire et de la Théorie des Langues Latines", Moscou, Russie, Juin 24-25 (2008)
5. Cardey, S., Greenfield, P., Anantalapochai, R., Beddar, M., Cornally, T., Devitre, D., Jin, G., Mikati, Z., Renahy, J., Kampeera, W., Melian, C., Spaggiari, L., Vuitton, D.: Le projet LiSe "Linguistique, normes, traitement automatique des langues et Sécurité: du data et sense mining aux langues contrôlées", p. 10 (2010)
6. Cardey, S.: Traitement algorithmique de la grammaire normative du français pour une utilisation automatique et didactique, Thèse de Doctorat d'Etat, Université de Franche-Comté, France (June 1987)
7. Cardey, S., Greenfield, P., Anantalapochai, R., Beddar, M., Devitre, D., Jin, G.: Modelling of Multiple Target Machine Translation of Controlled Languages Based on Language Norms and Divergences. In: IEEE Computer Society (IEEE-Xplore and IEEE Computer Society (CSDL) Digital Libraries, Indexed through IET INSPEC, EI (Compendex) and Thomson ISI), Japan, pp. 322–329 (2008)
8. Cardey, S.: Le miroir des peuples: phraséologie et traduction Le français comme médiateur de la diversité culturelle et linguistique, Ministère de la culture et l'Ambassade de France en Thaïlande, Bangkok (December 2007)

DAnIEL: Language Independent Character-Based News Surveillance

Gaël Lejeune, Romain Brixtel, Antoine Doucet, and Nadine Lucas

GREYC, University of Caen Lower-Normandy
Boulevard du Maréchal Juin BP5186-14032 Caen Cedex, France
`firstname.lastname@unicaen.fr`

Abstract. This study aims at developing a news surveillance system able to address multilingual web corpora. As an example of a domain where multilingual capacity is crucial, we focus on Epidemic Surveillance. This task necessitates worldwide coverage of news in order to detect new events as quickly as possible, anywhere, whatever the language it is first reported in. In this study, text-genre is used rather than sentence analysis. The news-genre properties allow us to assess the thematic relevance of news, filtered with the help of a specialised lexicon that is automatically collected on Wikipedia. Afterwards, a more detailed analysis of text specific properties is applied to relevant documents to better characterize the epidemic event (i.e., which disease spreads where?). Results from 400 documents in each language demonstrate the interest of this multilingual approach with light resources. DAnIEL achieves an F_1 -measure score around 85%. Two issues are addressed: the first is morphology rich languages, e.g. Greek, Polish and Russian as compared to English. The second is event location detection as related to disease detection. This system provides a reliable alternative to the generic IE architecture that is constrained by the lack of numerous components in many languages.

1 Introduction

Information Extraction (IE) aims at extracting structured views from free text and particularly from newswires. The Web provides many news sources in a variety of languages, and for instance the European Media Monitor collects about 40,000 news reports in 43 languages each day¹.

This paper focuses on multilingual IE with light resources and uses as application the epidemiological Event Extraction from the Web, a subdomain of IE whose goal is to detect and extract health-related events from news to send alerts to health authorities [1]. Tapping a wealth of information sources makes it theoretically possible to quickly detect important epidemic events over the world [2]. A health authority will want to monitor information with emphasis on disease outbreaks [3]. Until now, several approaches have been reported for epidemic surveillance on the Web [4] from full human analysis [5], keyword analysis [6] and web mining [7]. Human analysis is supposed to be more precise but has a great cost; keyword analysis is cheaper but lacks precision.

¹ <http://emm.newsbrief.eu/overview.html>

To perform global epidemic surveillance, researchers are facing a challenging problem: the need to build efficient systems for multiple languages at a reasonable cost. The classic IE architecture is built for a given language first, with components for each linguistic layer at sentence level (morphology, syntax, semantics). It has proved its high efficiency for applications in some important languages [8,4]. But most of the components involved in classical IE chains need to be rebuilt for each new language [9]. At a time when a greater variety of languages is observed on the Web, the coverage problem is still unsolved.

The approach advocated here is designed to be as media dependent as possible and as language independent as possible. It relies on established text-genre properties to perform analysis of news discourse taking advantage of collective style, more specifically on repetition patterns at certain places in text [10]. Though the rationale is different, technically the method is similar to relation discovery in open information extraction on the Web [11]. It also uses light crawled resources. Furthermore, its algorithmic basis permits a quick processing of large collections of documents.

The paper is organised as follows. In Section 2, we provide an overview of the multilingual approaches in IE. In Section 3 we present a system called *Data Analysis for Information Extraction in any Language* (DAnIEL), a genre-based IE system designed for managing multilingual news. In Section 4, we describe the corpus collected for this experiment. In Section 5 we show results and we elaborate on some of the results obtained on this corpus. Lastly, we conclude with a few additional remarks in Section 6.

2 Related Work

Use of the generic IE chain [12] as a model requires numerous and diverse components for each language. Components corresponding to a new language must be gathered or constructed. Two systems relying primarily on English, PULS² [6] and BIOCASTER³ [3] are used as well-known examples of classic IE systems with good results in English. A major disadvantage arises, however, for the end-user wishing to process a genuine multilingual corpus such as news feed. For most languages, efficient components will be lacking [13]. In recent years, machine learning was successfully used to fill gaps when one can find sufficient training data in a language which has enough common properties with the new one [11].

However, in epidemic surveillance, there is need to cover even very scarce resource languages or even dialects without training data. In a multilingual setting, state-of-the-art systems are limited by the cumulative process of their language-by-language approach. The detection and appropriate analysis of the very first news relating to an epidemic event is crucial, but it may occur in any language: usually the first language of description is that of the (remote) place where the event was located. This is why a new hypothesis from recent studies on media

² <http://medusa.jrc.it/medisys/helsinkiedition/en/home.html>

³ <http://born.nii.ac.jp/>

rhetorical devices [10] was put to trial. It relies on what can be called either pragmatics, or genre properties related to news discourse.

3 Our System: DAnIEL

The DAnIEL system presents a full implementation of a discourse-level IE approach. It operates at text-level, because it exploits the global structure of news in a newswire, that is information ordering as defined in [10], as opposed to the usual analysis of sentence-level linguistic layers (morphology, syntax and semantics). Entries in the system are text news, with their title and text-body. The details of the model are not justified here, but the main points as far as implementation are concerned are defined. Character-based refers to the fact texts are handled as sequences of characters, rather than as sequences of words, in order to consider all types of languages, including those where the definition and delimitation of words is difficult. The sequences that are extracted are not key words but machine-tractable strings that are linked to their order of appearance in text, paragraph after paragraph. A special interest has been put on describing the overall system as well as evaluating each part of it. The aim of the process is to extract epidemic events from news feed, and express them in the reduced form of disease-location pairs (i.e., what disease occurs in what country). Time is also important but will not be explained here for lack of space.

The system description is split in five subsections. DAnIEL uses a small knowledge base (Section 3.1) and its processing pipeline contains four steps: news article segmentation (Section 3.2), motifs extraction (Section 3.3), event detection (Section 3.4) and event localization (Section 3.5)

3.1 Knowledge Base

DAnIEL uses implicit knowledge on news writing and reading. Information is displayed carefully in news. The rules that are useful here are that information is displayed at important places, called positions. In journalistic style, writers use common disease names, because all newspaper readers know them. Media style rules also say they will appear before more specialised words in a piece of news. Last, important information is repeated, probably twice. Some similar observations are stated in different studies based on pragmatics or statistical studies (estimation of positive adaptation), notably on proper names [14].

DAnIEL uses only light lexical resources automatically collected from Wikipedia with light human moderation to pinpoint information that can be used to fill databases. The lexicon contains disease common names and some geographical names (countries). The lexicon needed with text-genre-based IE is quite small: roughly hundreds of items instead of tens of thousands in IE systems [15]. Indeed, Web-extracted disease names prove useful for dealing quickly with new languages, even without the assistance of a native speaker.

3.2 Article Segmentation

The main algorithm relies on the type of article being processed. The segmentation is important: as the approach is style-driven, having good judgement about the way the text is constructed is crucial. Key positions are the beginning and the end of text. For analysing press articles, the system relies on the title and beginning (the topical head) and checks which elements are repeated at key positions in the text. Because the hypothesis needed to be tested first, a coarse simplification was made to handle text length. Table 1 shows the three types of text according to length and the text windows corresponding to the text highlighted positions. Repetition are looked for in : Head (title plus the first paragraph), Tail (last two paragraphs) and Body (news article minus the Head).

Table 1. Article segmentation with respect to their number of paragraphs

Article type (example)	#paragraphs	Segments compared
Short (dispatches, relating hot news)	3 and less	All paragraphs
Medium (regular articles, event evolution)	4 to 10	Head and Body
Long (analysis articles, less current events)	more than 10	Head and Tail

For medium and long articles, the system extracts the substrings repeated in Head plus Body and Head plus Tail. For short articles, repeated substrings are considered irrespective of their position.

3.3 Motifs Extraction

The system checks repetitions at given positions in text, mainly beginning and end of text or text sub-units (paragraphs). To achieve that, character level analysis is allowed by computing non-gapped character strings as described by Ukkonen under the name motifs [16]. The main ideas are given in this section to enumerate those motifs in one or more text. Those motifs are substrings patterns of text with the following characteristics :

repeated: motifs occur twice or more;

maximal: motifs cannot be expanded to the left (*left maximality*) nor to the right (*right maximality*) without lowering the frequency.

For example, the motifs found in the string HATTIVATTIAA are T, A and ATTI. TT is not a maximal pattern because it always occurs inside each occurrence of ATTI. In other words its right-context (the characters on the right of all the occurrences of TT) is always I and its left-context A. All of these motifs in a set of strings are enumerated using an augmented suffix array [17].

For two strings $\mathcal{S}_0 = \text{HATTIV}$ and $\mathcal{S}_1 = \text{ATTIAA}$, both string in Σ^* , Table 2 shows the augmented suffix array of $\mathcal{S} = \mathcal{S}_0.\$1.\mathcal{S}_1.\$0$. $\$0$ and $\$1$ are lexicographically lower than any character in Σ and $\$0 < \1 . Augmented suffix array consists in the list of suffixes sorted lexicographically of \mathcal{S} (*SA*) and the Longest Common Prefix (*LCP*) between suffixes two at a time consecutively in

Table 2. Augmented suffix array of $\mathcal{S} = \text{HATTIV}\$1\text{ATTIAA}\$0$

i	LCP_i	SA_i	$\mathcal{S}[SA_i] \dots \mathcal{S}[n]$
0	0	13	$\$0$
1	0	6	$\$1\text{ATTIAA}\0
2	1	12	$\text{A}\$0$
3	1	11	$\text{AA}\$0$
4	4	7	$\text{ATTIAA}\$0$
5	0	1	$\text{ATTIV}\$1\text{ATTIAA}\0
6	0	0	$\text{HATTIV}\$1\text{ATTIAA}\0
7	1	10	$\text{IAA}\$0$
8	0	4	$\text{IV}\$1\text{ATTIAA}\0
9	2	9	$\text{TIAA}\$0$
10	1	3	$\text{TIV}\$1\text{ATTIAA}\0
11	3	8	$\text{TTIAA}\$0$
12	0	2	$\text{TTIV}\$1\text{ATTIAA}\0
13	0	5	$\text{V}\$1\text{ATTIAA}\0

SA ($LCP_i = \text{lcp}(\mathcal{S}[SA_i] \dots \mathcal{S}[n-1], \mathcal{S}[SA_{i+1}] \dots \mathcal{S}[n-1])$ and $LCP_{n-1} = 0$, n the size of \mathcal{S}).

The LCP allows the detection of repetitions, for example, the substring **ATTI** occurs at the offsets (1,13) in \mathcal{S} according to LCP_4 in Table 2. The process enumerates all the repeated substrings by reading through LCP .

- $LCP_i < LCP_{i+1}$: *open* a potential motif occurring at the offset SA_{i+1}
- $LCP_i > LCP_{i+1}$: *close* motifs previously created
- $LCP_i = LCP_{i+1}$: *sustain* motifs with the offset SA_{i+1} where it occurs in \mathcal{S}

The maximal criterion is checked when a motif is closed during the enumeration process. Two different potential motifs are equivalent if the last character of these motifs occurs at the same positions. For example, **TTI** is equivalent to **ATTI** because the last characters of these two motifs occur at the offsets (4,10). In that case, **ATTI** is kept as a *maximal* motif because it is the longest of its equivalents. The others motifs **A** and **T** are maximal because their contexts are different according to their occurrences.

All repetitions across different strings are detected at the end of the enumeration by mapping the offsets in \mathcal{S} with those in \mathcal{S}_0 and \mathcal{S}_1 . SA and LCP are constructed in $O(n)$ time [17], the enumeration process is done in $O(k)$ time, with k defined as the number of motifs and $k < n$ [16].⁴

3.4 Event Detection

DAnIEL filters out motifs according to article segmentation rules as described in Table 1, and to the list of disease names as explained in Section 3.1. It keeps motifs that are substrings found in two different sub-units, typically Head and Tail, and matching at least with one disease name. This comes from the genre-related rules saying that an important topic is highlighted in news, that common names are used to catch the reader’s attention and that the topic is repeated.

⁴ The code for computing these motifs in a set of strings is provided in PYTHON at <http://code.google.com/p/py-rstr-max/>

More formally, let \mathcal{S}_0 and \mathcal{S}_1 be the Head and the Tail of a long article and $\mathcal{S}_2 \dots \mathcal{S}_{n+1}$ the n entries in a diseases knowledge base. The process enumerates repetitions on $\mathcal{S}_0 \dots \mathcal{S}_{n+1}$ (section 3.3) and keeps motifs that occurs in \mathcal{S}_0 , \mathcal{S}_1 and any $\mathcal{S}_{1 < i \leq n+1}$. A heuristic ratio is used to check if a motif matches an entry: for a motif m occurring in key positions and in an entry \mathcal{S}_i in the list of diseases: $\frac{\text{len}(m)}{\text{len}(\mathcal{S}_i)} \geq \theta$ with len the number of characters in m and \mathcal{S}_i . The value of θ is discussed in section 5.2. This proves especially useful for morphologically rich languages; the need for a morphological analyzer is thus avoided. If DAnIEL finds no motif that matches its knowledge base using the θ threshold, it considers the document contains no event and thus is not relevant.

3.5 Event Localization

An event is minimally defined as a relation between a disease name, highlighted by its position and a place name. Once again, journalists' fairly strict writing principles help DAnIEL localize events without sentence-level extraction patterns. When talking about an epidemic, location of the event can be an important topic of the news. The explicit place names are found in the same way disease names are found, with the help of a reduced list extracted from Wikipedia.

When a journalist does not mention explicitly any location in the document, it means that this information relates to the issuing place. Hence, when no location is found using repetition rules (as seen in Section 3.4) and the list of geographical names, the location of the event is assumed to be the country of issue of the source by default (i.e., the newspaper or news agency country).

4 Corpus

Since the method that is tested considers full news, including title and text-body, no shared corpus was available. A corpus was to be collected in various languages from the Web. News corpora for English and Russian were collected from Google News' health category. As this category existed neither for Polish nor Greek, corresponding documents were collected from major newspapers' health categories⁵. We resorted to such pre-filtered sources because they are available. Even in health categories, however, only 8% of documents contained epidemic events. This strategy thus permitted to collect a significant number of relevant documents at a reasonable cost.

For measuring precision and recall on document filtering and event characterization, native speakers of each language⁶ annotated sets of about 500 documents covering a 8 week period from November 2011 to January 2012.

The characteristics of the evaluation corpus are shown in Table 3. The main characteristic is the fact that the length (in paragraph or characters) may vary a

⁵ "Gazeta", "Gazeta polska", "Dziennik zwiazkowy", etc. for Polish. "ΕΘΝΟΣ", "Το Βήμα", "ΕΞΙΠΕΣ", etc. for Greek.

⁶ Eight professional translators who were never involved in DAnIEL.

Table 3. Characteristics of the corpus

	English	Greek	Polish	Russian
#documents (relevant)	475 (31)	390 (26)	390 (30)	426 (40)
#paragraphs	10419	4216	4986	3565
avg. \pm std.	21.9 \pm 8.89	13.8 \pm 10.22	12.82 \pm 9.34	8.37 \pm 8.33
#characters (10^6)	1.60	2.09	1.19	1.64
avg. \pm std.	3372 \pm 1796	5382 \pm 5001	3059 \pm 2032	3871 \pm 5902

lot from one document to another. Annotators had to judge if these documents were relevant for informing health authorities about infectious diseases. If they judged them relevant, they had to further give the disease and location. This annotated corpus and the full annotation guidelines are available online⁷. The corpus is freely available for the community for further experiments.

5 Results and Evaluation

This section first highlights the efficiency of the repetition rule at key positions to select relevant press articles. Then DANIEL is evaluated against annotators' judgements on the evaluation corpus. The program to run the experiments, written in PYTHON, processes 2000 documents in less than 15 seconds (2.4Ghz dual core processor, 2Gb RAM), which is compatible with on-line surveillance.

5.1 Global Results

It is difficult to measure recall and precision when large amount of documents are processed, here the ground truth is the set of documents independent annotators found relevant. The F-measure is calculated with $\beta = 1$.

Table 4. Document filtering: precision, recall, F_1 -measure for best θ

	English	Greek	Polish	Russian	cumulated corpora
θ	0.85	0.75	0.8	0.85	best θ per language
Precision	0.77	0.76	0.73	0.85	0.78
Recall	0.97	1.0	0.85	0.88	0.93
F_1 -measure	0.86	0.86	0.79	0.86	0.85

Table 4 shows that recall is slightly better than precision. In this table, a different θ ratio was used for each language. Tuning the best ratio for each language permitted DANIEL to achieve 0.78 in precision with a 0.93 recall, for the cumulated corpus. This is unexpected, because it was feared that choosing to use a small lexicon would impair recall more than precision. Indeed it is an important question for a system that relies on small resources: the system should not miss too many events, particularly for epidemic surveillance. Table 5 shows the extent to which DANIEL generates silence and the reasons for errors.

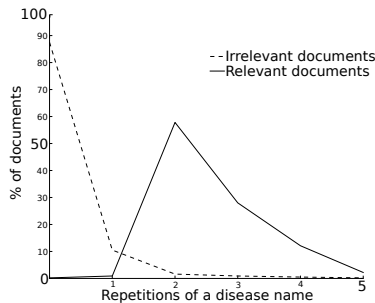
⁷ <https://lejeuneg.users.greyc.fr/daniel/>

Table 5. Document filtering, errors impairing recall

	English	Greek	Polish	Russian
#of relevant documents	31	26	30	40
Lack in lexicon	0	0	2	3
No repetition	1	0	1	1
Wrong matching	0	0	2	0
Silence	1	0	5	4

Errors due to the size of the lexicon are quite rare (5 are missed) and the repetition phenomenon is trustworthy: only three relevant documents were missed because no repetition matching with the disease name was found. More errors came from string recognition, because some diseases are referred by short names (in number of characters) and DAnIEL was unable to detect whether a disease was involved.

The news discourse model implemented through repetition rules at special positions efficiently selects relevant press articles on epidemiologic events. Figure 1 shows how frequent disease name repetition behaves in relevant articles (dotted line) and how rare it is in irrelevant ones (continuous line). This shows how this simple rule truly helps filter documents out: 97% of irrelevant and only 0.7% of relevant articles contained no repetition.

**Fig. 1.** Repetitions of disease name in relevant and irrelevant articles

5.2 Detailed Evaluation

Segmentation Filtering. The news segmentation described in Section 3.2 is intended to filter out uninteresting motifs. Table 6 shows the impact of this filtering in the total number of motifs.

Filtering Relevant Documents. In order to ponder the different features of our system, Table 7 shows the performance of two baselines: B1 relies on the presence of a disease name in the document and B2 relies on the repetition of the disease name. B1 highlights the problems one can encounter with morphologically rich languages because of the exact matching needed for the disease name. B2 shows the improvement in precision with the use of repetitions. Both baselines ignore the position criterion (Section 3.2) with $\theta = 1$ (Section 3.4).

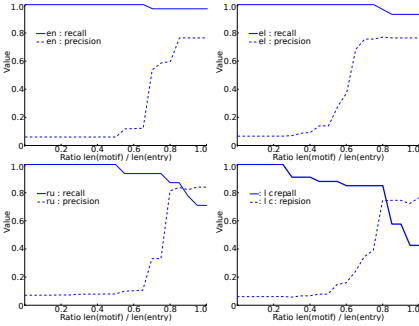
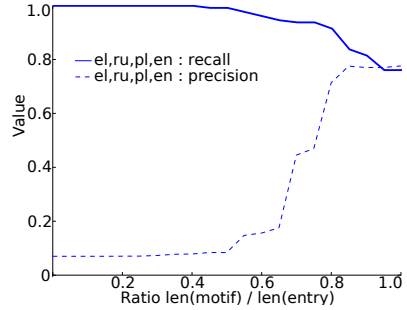
Table 6. Assessment of filtering impact, number of motifs for medium and long articles

	English	Greek	Polish	Russian
#documents	396	159	192	90
#motifs without segmentation (avg.)	1101.45	1242.81	1128.12	1311.07
#motifs with segmentation (avg.)	114.67	143.33	129.05	159.72
Filtering rate	9.60	8.67	8.74	8.20

Table 7. Evaluation of two baselines: precision, recall and F_1 -measure

		English	Greek	Polish	Russian	All
Baseline 1 (B1)	Precision	0.17	0.69	0.33	0.59	0.43
	Recall	1.0	0.62	0.79	0.61	0.68
	F_1 -measure	0.29	0.65	0.47	0.60	0.53
Baseline 2 (B2)	Precision	0.33	0.76	0.48	0.74	0.63
	Recall	0.97	0.45	0.56	0.61	0.60
	F_1 -measure	0.49	0.57	0.52	0.67	0.61

Evaluating the Threshold $\frac{\text{len}(\text{motif})}{\text{len}(\text{entry})} \geq \theta$. Figure 3 shows the results of empirical experiments to determine the appropriate string matching ratio between motifs extracted and knowledge base entry. $\theta = \frac{4}{5}$ is a good empirical value for processing the four different languages simultaneously and it might be optimized individually for each language (for example, 0.85 in Russian (Figure 2)).

**Fig. 2.** Recall and precision according to θ (English, Greek, Russian and Polish)**Fig. 3.** Recall and precision according to θ (all languages)

Event Localization. A large corpus (2000 documents for each language) was processed by DANIEL. Then, a subcorpus of relevant documents without explicit location was extracted. Those documents have been checked to assess if linking the events they describe to the source location is acceptable (Table 8).

In this corpus, roughly 70% of epidemic events contained an explicit location. Therefore results obtained show that the “implicit location” rule is efficient. For instance in Russian, among the 22% of documents where no location is explicitly mentioned, 78% are accurately localized with this simple rule. That leaves only 4.8% of all events incorrectly localized in Russian news.

Table 8. Performance of the implicit location rule

	English	Greek	Polish	Russian
# documents DAnIEL found relevant	93	188	213	230
# relevants documents without explicit location	46	33	35	51
Location = source	78.3%	81.8%	82.9%	78.4%
Location \neq source	21.7%	18.2%	17.1%	21.6%
Overall errors	12.2%	3%	2.8%	4.8%

Table 9. Evaluation by unique event

	Unique events	Detected	Missed
English	15	14	1 (6,6%)
Greek	17	17	0 (0%)
Polish	28	26	2 (7,1%)
Russian	23	21	2 (8,6%)
Total	57	54	3 (5,2%)

Level of Evaluation Unit: Document or Event? Evaluation per document is not necessarily adequate, when one considers a typical use case [18]. One can detect 99 documents describing the same event (e.g., flu in Spain) but miss an event because it was contained in only one document (e.g., Ebola in Congo). This should not be valued at 0.99 recall for the end-user. To evaluate how DAnIEL performs with respect to events rather than documents, further event-based annotations are made. Each disease-location pair (flu in Spain for instance) was considered as a unique epidemiological event regardless of the number of documents it has been reported in, over a 8 week time window.

Table 9 shows results of this experiment, demonstrating that only few full-fledged events (3 among 57) were missed. The system takes advantage of the fact that it has coverage in more than one language to detect events [19]. For instance, an event missed in Polish had been detected in Russian. Note that the total number of unique events in Table 9 is not the sum of unique events in reports, since a single epidemiological event can be reported in several languages.

6 Conclusion

The principles of a genre-based information extraction system called DAnIEL have been tested with success on English, Greek, Polish and Russian news. The system relies on very light, easy to get resources, and it is intended to help health authorities get precious information about on going infectious diseases spreading all around the world. In order to be multilingual, it uses genre related features and relies on text-style, specifically carefully selected types of string repetitions, rather than on sentence-level words or patterns specific to one or few languages.

The algorithm is based on the way news articles are rhetorically constructed. The detection of string repetitions permits to limit the number of components needed for monitoring new languages. No local analysis is used and a limited-size lexicon is enough. Experiments showed that the system might lack in precision, but has good recall (0.97 for English, 0.92 for the whole corpus). DAnIEL is

efficient at distinguishing relevant and really irrelevant documents, which makes it useful to filter large corpora, even with less known languages.

With an average F_1 -measure of 0.85 with appropriate tuning, DANIEL scores are below state-of-the-art systems like PULS or BIOCASTER, which are closer to 0.9 on English and a few other languages. But the resources that these systems need (lexicon, language parser, ontologies) are much more extensive and costly.

When no classical IE system is available or training data are too scarce, a text genre-based IE system can fill the gap efficiently. It can save efforts to filter relevant documents to be thoroughly parsed by existing techniques with high precision on major languages. In order to help IE research, the corpora used for this experiment are available to the community with annotations detached from original urls. It will be of interest for morphologically rich languages.

References

1. Linge, J., Steinberger, R., Weber, T., Yangarber, R., van der Goot, E., Al Khudhairy, D., Stilianakis, N.: Internet surveillance systems for early alerting of threats. *Eurosurveillance* 14(13) (2009)
2. Lyon, A., Nunn, M., Grossel, G., Burgman, M.: Comparison of web-based biosecurity intelligence systems: BioCaster, EpiSPIDER and HealthMap. *Transboundary and Emerging Diseases* (2011)
3. Son, D., Quoc, H.N., Ai, K., Collier, N.: Global health monitor - a web-based system for detecting and mapping infectious diseases. In: *International Joint Conference on Natural Language Processing*, pp. 951–956 (2008)
4. Hartley, D.M., Nelson, N.P., Walters, R., Arthur, R., Yangarber, R., Madoff, L., Linge, J., Mawudeku, A., Collier, N., Bronstein, J.S., Thinus, G., Lightfoot, N.: The landscape of international event-based biosurveillance. *Emerging Health Threats Journal* 3(e3) (2010)
5. Reilly, A.R., Iarocci, E.A., Jung, C.M., Hartley, D.M., Nelson, N.P.: Indications and warning of pandemic influenza compared to seasonal influenza. *Advances in Disease Surveillance* 5, 190 (2008)
6. Steinberger, R., Fuart, F., van der Goot, E., Best, C., von Etter, P., Yangarber, R.: Text mining from the web for medical intelligence. In: *Mining Massive Data Sets for Security*, pp. 295–310. OIS Press (2008)
7. Huttunen, S., Arto, V., von Etter, P., Yangarber, R.: Relevance prediction in information extraction using discourse and lexical features. In: *Nordic Conference on Computational Linguistics, Nodalida 2011*, pp. 114–121 (2011)
8. Ji, H.: Challenges from information extraction to information fusion. In: *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 507–515 (2010)
9. Du, M., Von Etter, P., Kopotev, M., Novikov, M., Tarbeeva, N., Yangarber, R.: Building Support Tools for Russian-Language Information Extraction. In: Habernal, I., Matoušek, V. (eds.) *TSD 2011. LNCS*, vol. 6836, pp. 380–387. Springer, Heidelberg (2011)
10. Lucas, N.: Stylistic devices in the news, as related to topic recognition. In: Kwiatkowska, A. (ed.) *Texts and Minds: Papers in Cognitive Poetics and Rhetoric*. Łódź, *Studies in language*. Peter Lang, Frankfurt am Main, vol. 26, pp. 301–316 (2012)

11. Etzioni, O., Fader, A., Christensen, J., Soderland, S.: Open information extraction: The second generation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 3–10 (2011)
12. Hobbs, J.R.: The generic information extraction system. In: Proceedings of the 5th Conference on Message Understanding, MUC5 1993, pp. 87–91. Association for Computational Linguistics, Stroudsburg (1993)
13. Steinberger, R.: A survey of methods to ease the development of highly multilingual text mining applications. *Language Resources and Evaluation*, 1–22 (2011)
14. Church, K.: Empirical estimates of adaptation: the chance of two Noriegas is closer to $\frac{p}{2}$ than p^2 . In: Proceedings of the 18th Conference on Computational Linguistics, vol. 1, pp. 173–179. Association for Computational Linguistics (2000)
15. Collier, N., Ai, K., Jin, L., et al.: A multilingual ontology for infectious disease surveillance: rationale, design and challenges. *Journal of Language Resources and Evaluation*, 405–413 (2007)
16. Ukkonen, E.: Maximal and minimal representations of gapped and non-gapped motifs of a string. *Theorie in Computer Science* 410(43), 4341–4349 (2009)
17. Kärkkäinen, J., Sanders, P., Burkhardt, S.: Linear work suffix array construction. *Journal of the ACM* 53(6), 918–936 (2006)
18. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 789–797 (2010)
19. Piskorski, J., Belyaeva, J., Atkinson, M.: On refining real-time multilingual news event extraction through deployment of cross-lingual information fusion techniques. In: Proceedings of European Intelligence and Security Informatics Conference (EISIC), pp. 38–45 (2011)

OOV Term Translation, Context Information and Definition Extraction Based on OOV Term Type Prediction

Jian Qu, Akira Shimazu, and Le Minh Nguyen

School of Information Science, JAIST
Nomi, Ishikawa, Japan
{qujian, shimazu, nguyenml}@jaist.ac.jp

Abstract. Although there are many existing approaches for solving the OOV term translation problems, but existing approaches are not able to handle different types of OOV terms, especially hybrid translations, such as “Kenny-Caffey syndrome (Kenny-Caffey氏症候群)”. We proposed a novel integrated ranking approach to consider the types of OOV terms before translating them. Thus, different types of OOV terms could be translated differently. Furthermore, the translations mined in other languages are also OOV terms, none of existing approaches offer the context information or definitions of the OOV terms. Users without special knowledge cannot easily understand meanings of the OOV terms. Our integrated ranking approach also extracts monolingual definitions and multilingual context information of OOV terms. Moreover, we propose a novel adaptive rules approach with Bayesian net and Adaboost for handling hybrid translations. Experiments show our approach performs better than existing approaches.

Keywords: Term translation, multilingual information retrieval.

1 Introduction

Out of vocabulary (OOV) terms are typically new terms that cannot be found in dictionaries. OOV terms can be classified into two groups; they are name type OOV terms such as personal names, place names, brands, etc. and technical type OOV terms such as new technical terms and new biomedical terms etc. Given an OOV term in source language, OOV term translation extraction aims to find the correct translation in target language.

Many existing approaches had explored various ways of finding translations for name type OOV terms in other languages. Biomedical type OOV terms, especially hybrid translations have received little attention in the past years. Hybrid translations use part target language and part source language. For example: a biomedical English OOV term “Kenny-Caffey syndrome” with its Chinese translation “Kenny-Caffey 症候群”, “Kenny-Caffey” is source language and “症候群” is target language. Another fundamental problem for existing approaches is the translations obtained in other languages are also OOV terms. These translations provided little information to users

without special knowledge. For example, "Mae West", its Chinese translation "梅蕙絲" does not offer any knowledge to users. Whether "梅蕙絲" is a company, a person or a Brand usually requires users to do further search.

In order to address the above problems, we propose to translate name type OOV terms and biomedical type OOV terms using different approaches. Thus, we will predict the type of OOV terms before translating them. We propose a novel integrated ranking approach to automatically predict the types of OOV terms. Then a novel adaptive rules approach together with supervised machine learning by Bayesian net with Adaboost is used for finding translations for biomedical type OOV terms, and we employ ranking list approach for finding translations for name type OOV terms. Furthermore our novel integrated ranking approach also extracts context information and definitions for name type and biomedical type OOV terms. For example, "Mae West", this approach would extract its Chinese translation "梅蕙絲", multilingual context information in English (*American/ actress/ playwright/ screenwriter/ writer*) and Chinese (*美国人/女演员/ 剧作家/ 编剧家/ 作家*) and its monolingual definition (*Mae West (born Mary Jane West on August 17, 1893 – November 22, 1980) was an American actress, playwright, screenwriter and sex symbol whose entertainment career...*)¹.

The remaining of this paper is organized as follows: Section 2 introduces related works. Our approach is overviewed in Section 3. In Section 4, we discuss the experiments and the results. And finally in Section 5, we conclude this paper and discuss the future works.

2 Related Works

Many researches in the past have proved automatic web mining is the most efficient approach for translating OOV terms [1-3]. Most OOV terms have their correspondent human translations nearby on the Internet [2, 3]. The translations mined from the Internet are usually high quality and require low man power cost. Automatic mining include three major steps, they are: 1) web retrieval aims to collect the snippets containing the possible translations of the OOV term from the Internet; 2) translation extraction aims to find the boundary of the translations in the snippets; and 3) translation selection aims to choose the correct translation from the extracted translations.

Many researchers in the past have endeavored to solve the OOV term translation problems with a similar translation extraction approach from Zhang and Vines. [1, 2, 4-7]. Zhang and Vines extract up to 30 Chinese characters before and after the OOV term when English OOV term is found. Then they use brute force translation extraction to generate all possible substrings of the extracted Chinese characters. This approach has a very high recall but may generate many noises, and it is difficult to handle biomedical type OOV terms, especially hybrid translations, since only Chinese characters are extracted.

Translation selection can be generally categorized into statistical based approaches and machine learning approaches. Ranking list approach from Zhang and Vines is a typical statistical based approach [4]. The ranking list approach uses lengths and frequencies of translation candidates to select the correct translation. In addition to improve the ranking list approach, Cheng *et al.* suggested the Symmetrical Conditional

¹ Web retrieved definition.

Probability and context dependency (SCPCD) and Lu *et al.* used the Symmetrical Conditional Probability (SCP) for selecting the correct translation [2, 8]. Machine learning based approaches for translation selection was proposed by Tifin *et al.* [9]. Many machine learning approaches for translation selection utilize support vector machine (SVM) [7]. However, parameters are very important and expensive for SVM.

Existing approaches have a large drawback on hybrid translations, for example “Kenny-Caffey syndrome” (Human: Kenny-Caffey症候群) (existing approach: 症候群). If “症候群” is applied to CLIR, many disease documents unrelated to “Kenny-Caffey syndrome” will be retrieved, because many Chinese medical terms end with the term “症候群”. Another fundamental problem of existing approach is the translations mined in other languages are also OOV terms, none of existing approaches offer the context information or definitions of OOV terms. Users without special knowledge cannot easily understand the detail meanings or the general ideas of OOV terms. It is very difficult for users without special knowledge to understand the meaning of a biomedical OOV term from only translations. Table 1 shows examples of translation results from our approach and existing approach. Seen from the table, our approach provides user the multilingual context information, which helps to explain the meaning of OOV terms.

Table 1. Examples of OOV term translations

OOV	Existing approach[4]	Our approach	
	Translation	Translation	Context information
Hereditary epidermolysis bullosa	先天性水皰症	先天性鬆懈水皰	skin disease/皮肤病
Leigh disease	萊氏症候群	萊氏症候群	rare neurometabolic disorder/ 罕见的神经代谢障碍

3 OOV Term Translation, Context Information and Definition Extraction Based on OOV Term Types

Existing approaches are very successful on name type OOV terms. However, they have many drawbacks on technical type OOV terms, especially hybrid translations from biomedical type OOV terms. In this paper, we developed a new adaptive rules approach for hybrid translations. However, this approach has some drawbacks on name type OOV terms.

In order to address the above problems, we propose to translate the OOV terms based on predication of types of OOV terms. Our approach takes into account a novel factor of different types of OOV terms. Thus different type OOV terms could be translated using different approaches. Existing ranking list approach is employed for translating name type OOV terms, and our novel adaptive rules approach is used for biomedical type OOV terms. A flow chart of our approach is shown in Fig. 1.

Seen from the figure, Our approach is developed into 5 steps, they are: 1) The snippet retrieval: documents (snippets) in both Chinese and English Languages containing OOV term and its possible translation, context information and definition are retrieved by

querying the English OOV terms over the Internet; 2) The snippet ranking and definition selection: English language snippets are ranked and OOV term definition are selected by our novel integrated ranking approach; 3) The multilingual context information extraction and OOV type predication: ontology tree is constructed from Word Net to extract the context information and predict the type of OOV terms; 4) The name type OOV term translation: name type OOV terms are translated by existing ranking list approach. 5) The biomedical type OOV term translation: biomedical type OOV terms are translated by our novel adaptive rules approach with Bayesian net and Adaboost.

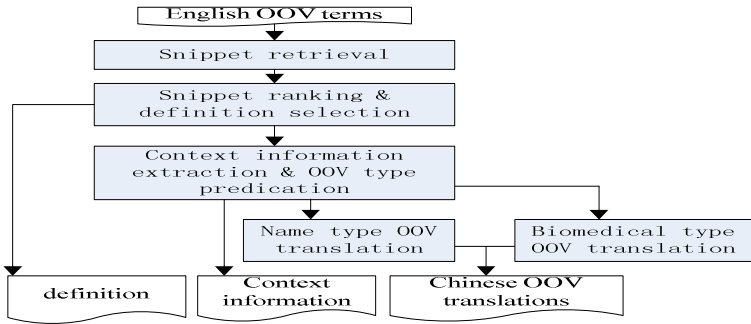


Fig. 1. Flow chart of our proposed approach

3.1 Snippet Retrieval

We feed English OOV terms to Bing Search API¹² and retrieve snippets from English language. We also retrieve snippets from Chinese language by limiting the retrieval language to Chinese. An example of English snippet containing the OOV term and its definition, and an example of Chinese snippet containing the OOV term and its translation are shown in Fig. 2.

English snippets:

Mae West - Wikipedia, the free encyclopedia (Title)
 Mae West (born Mary Jane West on August 17, 1893 – November 22, 1980) was an American actress, playwright, screenwriter and sex symbol whose ... (Summary)
http://en.wikipedia.org/wiki/Mae_West (URL)

Chinese Snippets:

α 1-抗胰蛋白酶缺乏症|症状|治疗 (Title)
 2009年1月10日 ... α 1-抗胰蛋白酶缺乏症(α 1-antitrypsin deficiency)是以婴儿期出现胆汁 ... 的糖蛋白, 在化学组成上与正常 α 1-AT 的区别是缺乏唾液酸基和糖基。 ... (Summary)
[www.yongyao.net/jbhtml/ \$\alpha\$ 1-kydbmqfz.htm](http://www.yongyao.net/jbhtml/α1-kydbmqfz.htm) (URL)

Fig. 2. An example of web retrieved snippets

3.2 Snippet Ranking and Definition Selection

In order to predict the types of OOV terms, we need to firstly extract the definitions of OOV terms. According to our observation, many OOV terms have their definitions on the web retrieved snippets. There are three main factors which can help us to identify the snippet with definitions. They are: 1) The snippets with definition possess some patterns. For example, "Mae West ... was an American actress ..." Some verbs can be

² <http://www.bing.com/toolbox/bingdeveloper>

used to identify snippets with definitions. 2) Search Engine rank the webpage very carefully, high Search Engine rank can also led to snippets with definition. 3) Web-pages with definitions are usually well organized organization, government, or educational web-pages. Thus domain names of the web-pages can also led to snippets with definitions. Combine above three factors, we propose a novel integrated ranking approach to rank the snippets with definitions. This approach takes snippets retrieved from the Internet and a list of verbs that can identify the snippets with definitions. We consider the locations and co-occurrences between the verbs and the OOV terms. Then we combine the domain ranking and Search Engine ranking to select the snippets with definitions. Domain ranks are given as follows: gov/org/edu/int > com/pro/net/info/ > else.

The integrated ranking is developed as follows. Let S_s be the summaries of English snippets retrieved from the Internet, S_t be the titles of English snippets retrieved from the Internet, OOV be the source OOV terms, V be the verb list, SR be the rankings from Search Engine, and DR be the domain ranking. For each OOV term, if the OOV term and a verb in V are both found in the S_s , we give it a rank 1, if the OOV term is found in the S_t , the sub-word of the OOV term and a verb in V are found in the S_s , we give it a rank 2; if only sub-word of the OOV term is found in the S_s , we give it a rank 3; if no sub-word of the OOV term is found in the S_s , we give it a rank 4.

After the ranks are assigned to the snippets, for each OOV term, we select one snippet with the highest snippet rank, the highest Search Engine rank and the highest domain rank. For each selected snippet, we extract the summary of the snippet as the definition of the OOV term. A detailed algorithm of the integrated ranking is shown below.

Algorithm snippets ranking and selection

Input: Summaries of snippets retrieved from the Internet, S_s ; Titles of snippets retrieved from the Internet, S_t ; OOV terms, OOV; Verb list, V ; Rankings from Search Engine, SR ; Disambiguation noun list, N ; Domain ranking list, DR .

Output: Snippets with ranking, SwR ; Selected Snippets, SwD .

```

For each OOV do
  If (OOV and V found in  $S_s$ )then
     $SwR = 1$ ,
  Else If (OOV found in  $S_t$ , sub-word OOV and V
  found in  $S_s$ )then
     $SwR = 2$ ,
  Else If (sub-word OOV found in  $S_s$ )then
     $SwR = 3$ ,
  Else
     $SwR = 4$ ,
 $SwD = MaxRank(SwR) \&\& MaxRank( SR) \&\& MaxRank( DR)$ 
End

```

An example of integrated ranking is shown in Table 2. As can be seen from this example, the snippet containing the correct definition of the OOV term "Mae West" gained a high rank.

Table 2. Ranking results of Snippets

URL	Title	Summary	Ranks(SWR, SR, DR)
http://en.wikipedia.org/wiki/Mae_West	Mae West - the free encyclopedia	Mae West (born Mary Jane West on August 17, 1893 – November 22, 1980) was an American actress, playwright, screenwriter and sex symbol whose entertainment career ...	1,1,1
http://www.imdb.com/name/nm0922213/	Mae West - IMDb	My Little Chickadee (1940) · Klondike Annie (1936). Mae West was born in Brooklyn, New York, to ...Soundtrack: I'm No Angel (1933) ...	1,2,2
http://www.youtube.com/watch?v=qVrfHXnUJFc	Mae West in I'm No Angel Trailer - YouTube	With Cary Grant in this 1933 comedy classic. Fortuneteller: I see a man in your life. Mae: What, only one?...	2,3,2

3.3 Multilingual Context Information Extraction and OOV Type Prediction

For each OOV term with its definition from the previous step, we extract the context information from the OOV term. We construct two ontology trees from Word Net to predict the type of OOV terms. Word net is a large English lexicon [10]. We use 7 Word Net super node noun terms (such as country, occupation, industry, etc) and extract their brief hyponyms to construct the ontology tree for name type OOV terms. Then we use 5 Word Net super node noun terms (such as illness,) and extract their brief hyponyms to construct the ontology tree for biomedical type OOV terms. We use the definition of each OOV term to search against the ontology trees. When any word in definition is found in the ontology trees, we extract such word as a context information of the OOV term. Then, we predict this OOV term to either name type or biomedical type according to the majority number of context information found in different ontology trees. Furthermore, these context information are then translated into Chinese by multilingual dictionary³.

3.4 Translation Extraction and Selection for Name Type OOV Terms

After the types of OOV terms are predicted, the name type OOV terms are translated by existing ranking list approach [4]. For each OOV term, we select the top ranked candidates as the Chinese translation.

3.5 Translation Extraction and Selection for Biomedical Type OOV Terms

We propose a new adaptive rules approach for biomedical type OOV terms in order to handle hybrid translations. The translation extraction and selection for biomedical type OOV term is developed into 3 steps, they are: firstly, the translation extraction using novel adaptive rules approach; secondly, the feature extraction, and finally the translation selection using Bayesian net with Adaboost.

Translation Extraction for Biomedical Type OOV Terms

According to our observation, we found out some hybrid translations of the OOV terms may not only use the target language alphabets or characters, but also use some alphabets,

³ <http://www.oxfordlanguagedictionaries.com/Public/PublicHome.html>

characters or symbols from the source language. We propose to include the alphabets, characters or symbols of the source language by using an adaptive rules approach.

The adaptive rules approach uses a set of predefined regular expression matching rules as the base rules. The base rules are modified by each OOV term to form the adapted regular expression matching rules for translation extraction.

The adaptive rules approach is developed as follows. Let S_n be the snippets retrieved from the Internet, OOV be the source OOV terms, A be any alphabets, characters or symbols, Ac be any Chinese characters, Re be regular expression matching rules, Ar be adapted matching rules, and Tc be Chinese translation candidates. For each OOV term, if it is found in the snippets, we add the substring of the OOV term to the regular expression matching rules to create the adapted matching rules. Then we scan for the nearest Chinese character in front of or after the OOV terms. Once we find the Chinese character, we try to match the string around the Chinese character with the adapted matching rules. If there are one or more rules that match the string, we extract the matched parts of the string as the Chinese translation candidates. The detailed algorithm of this approach is explained below. An example of Re and Ar for OOV term "Kenny-Caffey syndrome" is shown in table 3.

Algorithm Translation extraction

```

Input: Snippets retrieved from the Internet  $S_n$ , OOV terms OOV, Any alphabets, characters or symbols A, Any Chinese characters Ac, Regular expression matching rules Re,

Output: Adapted matching rules Ar, Chinese Translation candidates Tc,

For each OOV found in  $S_n$  do
Ar = Re + Substring OOV,
  If (Ac found in front or behind OOV) then
    Continue;
  If (Ar found near Ac+A) then
    Matching Ar with Ac+A;
    Tc = Ac+A;
  End
End
End
    
```

Table 3. An example of Re and Ar

#	Re	Ar
1	a-z/Chinese characters	Kenny-Caffey/Chinese characters
...

Feature Extraction

In this subsection, we extracted totally 19 different features from translation candidates. These features include: average distances, co-occurrence distance, term frequencies, symmetric conditional probability (SCP), modified association measures, lengths of OOV and translation, and length similarity. We describe the details of these features as follows.

Distances between OOV and translation

The closer a translation candidate to its source OOV term the more likely that translation is correct [2]. Some translations occur both in front and after the OOV term, but some translations only occur in front or after the OOV term. To present the actual

locations between the OOV and translations, we need to consider the average distance $Dist(c_i e_i)$, average front distance $Dist(c_i, e_i)$ and the average back distance $Dist(e_i, c_i)$.

Co-occurrence distance

Co-occurrence distance ($CDist$) is the sum of average distance between OOV and translation candidate over the co-occur frequency between OOV and translation candidate. It is computed as follows.

$$CDist = \frac{sum(Dist(c_i e_i))}{tf(c_i e_i)} \quad (1)$$

A modification of the above feature ($CwDist$) was proposed by Zhang *et al.* [7], they use the web retrieved page count instead of the $tf(c_i e_i)$.

Equation (1) shows the calculation of $CDist$, where $tf(c_i e_i)$ is the co-occur frequency between OOV and translation candidate.

Term frequencies

We collect the term frequencies of the translation candidates $tf(c_i)$, OOV $tf(e_i)$, and the co-occur frequencies of translations and OOV $tf(c_i e_i)$. Furthermore, to cope with the average front distance and average back distances, we also collect the front frequency $tf(c_i, e_i)$ and back frequency $tf(e_i, c_i)$ for the translation candidates.

Symmetrical Conditional Probability

Symmetrical Conditional Probability (SCP) [2, 8, 11] checks each alphabet, character and substring in the possible translation to determine whether this translation is a term or a sentence.

Modified Association Measures

We propose the modified association measures, which do not require the total number of pages in the Internet. They take the webpage count of OOV terms $S(e_i)$, translations $S(c_i)$, the webpage count of OOV terms co-occur with translations $S(e_i \wedge c_i)$ and the webpage count of OOV terms occur without translations $S(e_i \wedge \neg c_i)$ from the Internet. These features utilize the Search Engines to remove some possible wrong translation candidates, because Search Engines use some predefined segmentation tools and sometimes hire human to eliminate the meaningless Chinese strings.

Support

$$Supp(e_i \rightarrow c_i) = S(e_i \wedge c_i) \quad (2)$$

Confidence

$$Conf(e_i \rightarrow c_i) = \frac{S(e_i \wedge c_i)}{S(e_i)} \quad (3)$$

Lift or Interestingness

$$lift(e_i \rightarrow c_i) = \frac{S(e_i \wedge c_i)}{S(e_i)S(c_i)} \quad (4)$$

Conviction

$$Conv(e_i \rightarrow c_i) = \frac{S(e_i)(\neg c_i)}{S(e_i \wedge \neg c_i)} \quad (5)$$

Equation (2) is the Support of association measure, equation (3) is the Confidence of the association measure, equation (4) is the Lift of the association measure, and equation (5) is the Conviction of the association measure. e_i is the OOV term, c_i is the translation and $S(e_i)$ is the number of pages returned by the Search Engine when e_i is submitted as a query. $(-c_i)$ is assumed to be 1, because translations of OOV terms have a very small portion when compare to the whole Internet.

Length of OOV and translation candidates

The translation of OOV should have similar ratio of length, we collect the lengths of translation candidates ($|c_i|$), lengths of OOV terms ($|e_i|$) and the differences between them $D(|e_i|, |c_i|)$.

Length similarity

We also employ the length similarity ratio from Shi [12], it is a normalized length difference.

Translation Selection for Biomedical Type OOV Terms

In this section, we explain our candidate selection approach. It is developed into two parts, the statistical filter, and the Bayesian net translation selection with Adaboost.

One OOV term can retrieve up to few hundreds of translation candidates, most of them are substrings of the correct translation, and some of them are the longer strings of the correct translation. Two features in our feature set can simply filter some wrong candidates, they are co-occur frequency ($tf(c_i e_i)$) and location distance ($Dist(c_i e_i)$). Both features are very important to the candidate selection, if a Chinese translation co-occurs very often with the source English OOV term and this translation is found very close to the source English OOV term, then this translation may less likely be the wrong translation (noise). Our filter takes the top 70% of the co-occur frequency and the shortest location distance between OOV and the translations. A recall test was performed to evaluate the setting of this filter, the result is shown in Fig. 3.

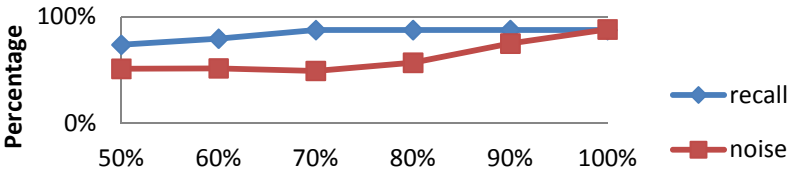


Fig. 3. Recalls of the filter

To handle the diversity of OOV terms, we employed the Bayesian net which can establish an inference reasoning and causality between OOV terms and the translation candidates. We also employ the Meta level-Adaboost to handle the over fitting problems [13].

4 Experiments and Discussion

In this section, we describe our data collection, experiments and discuss the results.

4.1 Data Collection

We collected English name type OOV terms from lists of top 500 companies⁴, famous people⁵, car brands⁶, CPU code names⁷. Furthermore, we collected English biomedical terms from International Classification of Diseases, Functioning, and Disability version 9 (ICD9)⁸. We randomly selected 10% from each above list. Combining the above lists, we obtained a total of 203 OOV terms, 76 of them are name type OOV terms and rest are biomedical type OOV terms. The idea of this data collection is to test the flexibility and possibility of our approach on both name type and biomedical type OOV terms. In order to create a baseline, we hired two Chinese graduate students manually search the web/dictionary to find the correct OOV term definition, context information and Chinese translation.

We retrieved a total of 7,182 English snippets and 5,897 Chinese snippets after querying the above OOV terms to the web. Then we process these snippets with our integrated ranking approach.

4.2 Experimental Results for OOV Type Predication, Definition and Multilingual Context Information Extraction

Although the input were 203 OOV terms, only 201 were able to retrieve form the Internet. Our novel integrated ranking approach correctly predicted 75 name type OOV terms, and 118 biomedical type OOV terms. Furthermore, our integrated ranking approach also extracted the monolingual definition of the English OOV term; the monolingual context information of the English OOV term; and the Chinese translations of the context information. We compared each OOV term with the baseline to check for correctness. The detailed results are shown in Table 4.

Table 4. Experimental results for OOV type prediction, definition and context information extraction

Baselines	OOV type prediction			monolingual		multilingual
	precision	recall	accuracy	Defini- tion(accuracy)	context information(accuracy)	Chinese context infor- mation(accuracy)
All OOV(201)	95.31%	96.54%	96.02%	170(84.58%)	181(90.05%)	179(89.05%)
Name type OOV(76)	91.46%	98.68%	96.02%	74(97.37%)	75(98.68%)	74(97.37%)
Biomedical type OOV(125)	99.16%	94.40%	96.02%	96(78.80%)	106(84.80%)	105(84.00%)

4.3 Experimental Setup for OOV Term Translation

After we predicted the types of OOV terms, the name type OOV terms are translated using ranking list approach and biomedical type OOV terms are translated using our novel adaptive rules and Bayesian net with Ada boost approach. RapidMiner [13] is used for

⁴ <http://money.cnn.com/magazines/fortune/fortune500/2011/index.html>

⁵ http://www.selfcreation.com/creation/famous_people.htm

⁶ http://wiki.answers.com/Q/List_of_car_brands

⁷ http://www.cpubenchmark.net/cpu_list.php

⁸ <http://www.cdc.gov/nchs/icd/icd9.html>

machine learning since it details out each results on the learning process. We used 10-fold cross validation to experiment on the annotated data collection. Although some OOV terms have few correct translations, we only select one for the final evaluation.

To compare our approach with existing approaches, we tested the same data set with the Pat-tree and SVM approach from Zhang *et al.* [7]. We also tested the ranking list approach from Zhang & Vines [4].

4.4 Experimental Results for OOV Term Translation

Table 5 shows the experimental results for OOV term translation. We can see that ranking list approach from Zhang & Vines achieved accuracies of 77.61% and 67.66% in translation extraction and translation selection respectively. While Pat-tree and SVM approach from Zhang *et al.* gained accuracies of 78.61% and 71.14% in translation extraction and translation selection respectively. Our proposed approach is significantly better than existing approaches. We gained accuracies of 93.04% and 87.56% in translation extraction and translation selection respectively.

Table 5. Comparison of our proposed approach with existing approaches

Approaches	Transliterated OOVs	Correct translation mined (Name type)(biomedical type)	Correct translation mined in OOV terms(Name type)(biomedical type){accuracy}	Correct translation selected in OOV terms(Name type)(biomedical type){accuracy}
Our approach	201	212(79)(125)	187(74)(113) {93.04%}	176(74)(102) {87.56%}
Pat-tree SVM approach		178(79)(99)	158(73)(85) {78.61%}	143(71)(72) {71.14%}
Ranking list approach		179(81)(98)	156(74)(82) {77.61%}	136(74)(62) {67.66%}

4.5 Discussions

Our approach performed better because we consider the types of OOV terms. Different approaches were used for different types of OOV terms. Furthermore, our adaptive rules approach can extract many translations where existing approaches failed to extract, mostly because we considered the existence of the hybrid translations. Moreover, our approach extracts the monolingual information and the multilingual context information for the OOV term. However, some drawbacks are within our integrated ranking approach. If Search Engine, our algorithm and domain ranking all ranked a wrong snippet highly relevant to an OOV term, we may get an error. For example, query term #144 "Fucosidosis"(a biomedical OOV term), however the snippet with high rank for "Fucosidosis" contains definition for a person named "Fucosidosis".

5 Conclusion

OOV term has always been a problem for natural language processing, especially for information retrieval. Although there are many existing approaches for solving the OOV term translation problems, but existing approaches are not able to handle different types of OOV terms. Furthermore, the translations mined in other languages are also OOV terms, none of existing approaches offer the context information or

definition of the OOV terms. Users without special knowledge cannot easily understand the detail meanings or the general ideas of the OOV terms. We proposed an integrated ranking approach for predicting the types of OOV terms and extracting the monolingual definitions and the multilingual context information. Moreover, we propose a novel adaptive rules approach with Bayesian net and Adaboost for handling hybrid translations. We evaluate our approach with both name type and biomedical type OOV terms. Our approach achieved high accuracies of 96.02% for OOV type prediction, 84.58% for monolingual definition extraction, and 89.06% for multilingual context information extraction. Our approach also achieved high accuracies of 93.04% in translation extraction and 87.56% in translation selection for OOV term translation. In future, we will develop better translation extraction approach, and improve our OOV context information extraction approach.

References

1. Lu, W.-H., Chien, L.-F., Lee, H.-J.: Anchor text mining for translation of Web queries: A transitive translation approach. *ACM Trans. Inf. Syst.* 22(2), 242–269 (2004)
2. Cheng, P.-J., et al.: Translating unknown queries with web corpora for cross-language information retrieval. In: *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 146–153. ACM, Sheffield (2004)
3. Zhang, Y., Huang, F., Vogel, S.: Mining translations of OOV terms from the web through cross-lingual query expansion. In: *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 669–670. ACM, Salvador (2005)
4. Zhang, Y., Vines, P.: Using the web for automated translation extraction in cross-language information retrieval. In: *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 162–169. ACM, Sheffield (2004)
5. Zhang, Y., Vines, P.: Detection and translation of OOV terms prior to query time. In: *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 524–525. ACM, Sheffield (2004)
6. Zhang, Y., Vines, P., Zobel, J.: Chinese OOV translation and post-translation query expansion in chinese-english cross-lingual information retrieval. *ACM Transactions on Asian Language Information Processing (TALIP)* 4(2), 57–77 (2005)
7. Zhang, Y., Wang, Y., Xue, X.: English-Chinese bi-directional OOV translation based on web mining and supervised learning. In: *ACL-IJCNLP 2009 Conference Short Papers*, pp. 129–132. Association for Computational Linguistics, Suntec (2009)
8. Lu, C., Xu, Y., Geva, S.: Translation disambiguation in web-based translation extraction for English-Chinese CLIR. In: *ACM Symposium on Applied Computing*, pp. 819–823. ACM, Seoul (2007)
9. Tiffin, N., et al.: Integration of text and data-mining using ontologies successfully selects disease gene candidates. *Nucleic Acids Res.* 33, 1544–1552 (2005)
10. Fellbaum, C.: *WordNet An Electronic Lexical Database* (1998)
11. Ferreira da Silva, J., Dias, G., Guilloré, S., Pereira Lopes, J.G.: Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In: Barahona, P., Alferes, J.J. (eds.) *EPIA 1999. LNCS (LNAI)*, vol. 1695, pp. 113–132. Springer, Heidelberg (1999)
12. Shi, L.: Mining OOV Translations from Mixed-Language Web Pages for Cross Language Information Retrieval. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., van Rijsbergen, K. (eds.) *ECIR 2010. LNCS*, vol. 5993, pp. 471–482. Springer, Heidelberg (2010)
13. Rapidminer, Rapidminer data mining tool (2009)

Exploiting a Web-Based Encyclopedia as a Knowledge Base for the Extraction of Multilingual Terminology

Fatiha Sadat

Université du Québec à Montréal
201 av. Président Kennedy,
Montréal, QC, H3X 2Y3 Canada
sadat.fatiha@uqam.ca

Abstract. Multilingual linguistic resources are usually constructed from parallel corpora, but since these corpora are available only for selected text domains and language pairs, the potential of other resources is being explored as well. This article seeks to explore and to exploit the idea of using multilingual web-based encyclopaedias such as Wikipedia as comparable corpora for bilingual terminology extraction. We propose an approach to extract terms and their translations from different types of Wikipedia link information and data. The next step will be using linguistic-based information to re-rank and filter the extracted term candidates in the target language. Preliminary evaluations using the combined statistics-based and linguistic-based approaches were applied on different pairs of languages including Japanese, French and English. These evaluations showed a real open improvement and a good quality of the extracted term candidates for building or enriching multilingual anthologies, dictionaries or feeding a cross-language information retrieval system with the related expansion terms of the source query.

Keywords: terminology, comparable corpora, translation, Cross-Language Information Retrieval, linguistics-based information.

1 Introduction

In recent years two types of multilingual corpora have been an object of studies and research related to natural language processing and information retrieval: parallel corpora and comparable corpora. The parallel corpora are made up of original texts and their translations (Morin et al., 2004 ; Véronis, 2000). This allows texts to be aligned and used in applications such as computer-aided translator training and machine translation systems. This method could be expensive for any pair of languages or even not applicable for some languages, which are characterized by few amounts of Web pages on the Web. On the other hand, non-aligned comparable corpora, more abundant and accessible resources than parallel corpora, have been given a special interest in bilingual terminology acquisition and lexical resources

enrichment (Dejean et al., 2002; Fung, 2000; Gœuriot et al., 2009a; Gœuriot et al., 2009b; Morin et al., 2006; Nakagawa et al., 2000; Rapp, 1999; Sadat et al., 2003; Sadat et al., 2004). Comparable corpora are defined as collections of texts from pairs or multiples of languages, which can be contrasted because of their common features in the topic, the domain, the authors, the time period, etc. Comparable corpora could be collected from downloading electronic copies of newspapers and articles, on the WWW for any specified domain.

Among the advantages of comparable corpora; their availability, consistency and utility for research on Natural Language Processing (NLP). In another hand, recent publications on bilingual terminology extraction from comparable corpora have shown promising results although most used comparable corpora are domain-specific, which causes limitations on the usage diversity, the domain and the quality of terminology.

This paper intends to bring solutions to the problem of lexical coverage of existing linguistic resources such as multilingual ontologies and dictionaries, but also to the improvement of the performance of Cross-Language Information Retrieval. The main contribution of the current study is an automatic acquisition of bilingual terminology from Wikipedia¹ articles in order to construct a bilingual ontology or enhance the coverage of existing ontologies.

The remainder of the present paper is organized as follows: Section 2 presents an overview of Wikipedia. Section 3 presents the different steps for the acquisition of bilingual terminology using a two-stage corpus-based translation model. Experiments and evaluations are related in Section 4. Section 5 concludes the present paper.

2 An Overview of Wikipedia

Wikipedia (*having the pronunciation wikipɛ'dʒa or vikipe'dʒa*) is an online multilingual encyclopedia based on the Internet, universal, multilingual and working on the concepts of a wiki, i.e. a web site with freely updatable web pages from all or a part of visitors of that site.

Wikipedia offers a gigantesque repository of multilingual data to exploit automatically for different aims in NLP. Different search engines such as *Google*² or *Yahoo*³ or Wikipedia's can be used for the implementation of the approach to extract bilingual terminology from comparable corpora and its related evaluations.

Wikipedia offers a neutral content that can be verified and updated freely by any editor. The edition of collaborative documents can be monolingual or multilingual. Actually, the French version of Wikipedia (francophone⁴) has more than 943 399 articles and more than 5 000 actives contributors⁵.

¹ <http://www.wikipedia.org>

² <http://www.google.com>

³ <http://www.yahoo.com>

⁴ <http://fr.wikipedia.org>

⁵ Information of April 30th 2010 at 9:34 am.

This considered linguistic resource can be used as parallel or comparable corpora: it can be considered as gigantesque lexical resource, available freely for all users, for many domains and diverse languages. However, its exploitation in NLP research is recent, not completely pertinent and still requires theoretical ideas and practice on its statute, characteristics and limits (Adafre et al., 2006; Schönhofen et al., 2007; Erdmann et al., 2008a; Erdmann et al., 2008b; Erdmann et al., 2009; Adar et al., 2009; Mohammadi et QasemAgharee, 2009 ; Yu et Tsujii, 2009a; Yu et Tsujii, 2009b).

The aim of the current study is the acquisition of bilingual or multilingual terminology from Wikipedia articles, which is automatic and language independent. The evaluation of our ideas and approach is done on different pairs of languages including French, English and Japanese.

According to figure Fig. 1., the number of Wikipedia articles for most of European languages, has achieved a limit that allows this resource to be used in NLP research and more specifically in multilingual information extraction and retrieval. Although, the advance of this resource content, most of studies were concentrated on the monolingual aspect (Voss 2005). We are interested in the multilingual aspect of Wikipedia in order to extract the pertinent terminology for the development or enrichment of a multilingual ontology or dictionary.

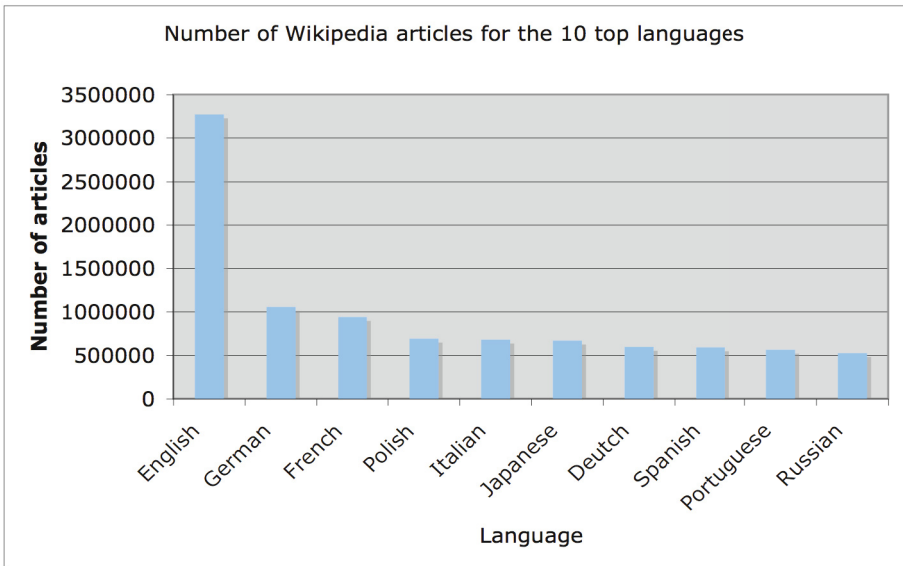


Fig. 1. Number of Wikipedia articles for the 10 most used languages

3 Multilingual Terminology Extraction

The process to extract bilingual terminology from Wikipedia documents is described as follows: (1) construction of comparable corpora; (2) translation using a statistical approach; (3) combination to linguistic information in order to filter and re-rank the extracted terminology as described in Sadat et al. (2003, 2004).

First, we consider a preliminary query Q of n words in a source language S to input in Wikipedia search engine. The resulted document is used as a first document for the corpus in the source language S . The usage of the inter-language link in the target language T for this document will lead to a corpus in a target language T .

Following this first step and exploiting the links in the same document as well as the inter-language links, comparable corpora are built for the query Q .

In this study, we use the term *deep* in the same document to define the number of times links in the same language are used in our corpora. Example, a first document $corp_1$ is described by $deep_0$; using the i links that are included in this first document will lead to $deep_1$ and to an extended corpus $corp_2$, using the j links that are included in the $corp_2$ will lead to $deep_2$ and to an extended corpus $corp_3$. This step can be terminated at $deep_m$ to result in a comparable corpus $corp_{m+1}$.

The exploitation of all links related to the set [$deep_0 \dots deep_m$] in the source language will lead to a corpus in this language. In another hand, a parallel exploitation of inter-language links in the target languages for the same set [$deep_0 \dots deep_m$] will lead to a corpus in the target language that is comparable to the one in the source language.

This approach can be used to build a multilingual corpus in several languages according to the availability of documents in all those languages in Wikipedia sites.

Second, the statistical phase will realize the alignment between terms of the source language and those in the target language in a bilingual way, which means two-by-two (or bilingual) extraction of terminology.

Considering the constructed comparable corpora from Wikipedia articles, we apply the following steps to extract the bilingual terminology:

1. *Extraction of terms from source and target languages documents:* In this step, terms with the following part of speech tags are extracted: *noun, verb, adverb, adjective*.
2. *Construction of context vectors in both languages:* For each term w , a context vector is constructed considering all terms that co-occur with the term w in a specified window size of one phrase. The mutual information (Dunning, 1993) is used as a co-occurrence tendency measure.
3. *Translation of the context vector content in the source language to the target language:* Context vectors of words in the source language are translated into the target language using the Wikipedia resource as translator. This step requires using the interlink information of Wikipedia for word translation. If needed, the *Wiktionaire*⁶ is used to overcome the limitations of Wikipedia and

⁶ <http://fr.wiktionary.org/>

to deal with out-of-vocabulary words. In the current study, we are interested by exploiting specifically Wikipedia as a multilingual and lexical resource, although it is possible to use a bilingual dictionary or a freely available machine translation to overcome the limitations of the translation.

4. *Construction of similarity vectors*: Context vectors (original and translated) of words in both languages are compared using the *cosine metrics*. Other measures such as the *Jaccard distance* or the *Dice coefficient* can be considered.

The third step consists on a *linguistics-based pruning approach*. Terms and their translations that are morphologically close enough, i.e., close or similar POS tags, are filtered and retained. We restricted the pruning technique to nouns, verbs, adjectives and adverbs, although other POS tags could be treated in similar way.

Finally, the generated translation alternatives are sorted in decreasing order by similarity values. Rank counts are assigned in increasing order, starting at 1 for the first sorted list item. A fixed number of top-ranked translation alternatives are selected and misleading candidates are discarded.

In this proposed approach, all monolingual links in a document are used to extract terms and concepts in the related language. In another hand, links involving two or several languages are used to retrieve terms across those languages.

4 Evaluations

Our preliminary evaluations using the proposed strategy were based on different pairs of languages including French, English and Japanese. Different sizes of the Wikipedia corpus were used and referenced here by the term *deep*.

Table 1 shows the size of the bilingual corpora according to the exploitation of same-language links and inter-language links.

Tables 2,3 and 4 show the results of the obtained bilingual terminology according to different sizes of the corpora for the French-English, Japanese-French and Japanese-English pairs of languages, respectively.

Note that we used a first query including the terms « *infection hospital illness tuberculosis* » which is a part of NTCIR-7⁷ test collection in CLIR, in the three languages, i.e. French, Japanese and English.

Table 5 shows an example for the extracted bilingual terminology in English for the source term « *santé* » in French (which means *health* in English) with *deep*₃.

The obtained terminology is very useful for building a bilingual ontology (or multilingual). The extracted terms have a certain semantic relationship with the source term and the resulted documents in the source and target languages can be exploited in order to define the semantic relations and thus build a multilingual ontology.

⁷ <http://aclia.lti.cs.cmu.edu/ntcir8>

Table 1. Sizes of Wikipedia corpora according to different links exploitation

Deep	Number of tokens/ articles in French	Number of tokens/ articles in English	Number of tokens/ articles in Japanese
0	388 / 4	510 / 4	57 / 4
1	4 511 / 61	5 633 / 51	185 / 10
3	161 967 / 2 634	205 023 / 2 121	10 964 / 266
7	533 931 / 9 077	657 035 / 7 110	45 378 / 977

Table 2. Examples of the extracted bilingual terminology (French-English) according to different sizes of the Wikipedia corpus

Deep	Source term (fr.)	Number of candidates (eng.)	Ideal translation (eng.)	Rank
0	organisation	14	Organization	1
	organisation	14	Institution	4
	organisation	14	compagny	8
	organisme	105	organism	4
	Maladie	101	Disease	14
	Santé	89	Health	1
	hôpital	19	Hospital	3
1	admission	90	admission	1
	algue	88	Algae	1
	thérapie	52	therapy	1
	animal	369	animal	2
	assistance	85	support	3
	blesure	269	injury	3
	épidémiologie	186	epidemiology	5
3	abeille	443	bee	1
	narcotique	1656	narcotic	1
	assurance	289	insurance	2
	chimie	2044	chemistry	3
	silicone	132	silicone	3
	médecine	416	medicine	4
	réanimation	1004	resuscitation	5
	taxonomie	1841	taxonomy	7

Table 3. Examples of the extracted bilingual terminology (Japanese-French) according to different sizes of the Wikipedia corpus

Deep	Source term (jap.)	Number of candidates (fr.)	Ideal translation (fr.)	Rank
0	感染	14	Infection	1
1	イングランド	16	Angleterre	2
	けっか	6	Résultat	1
	世界	14	Monde	1
3	アレルギー	236	Allergie	2
	セルロース	233	Cellulose	2
	ワクチン	102	Vaccin	1

Table 4. Examples of the extracted bilingual terminology (Japanese-English) according to different sizes of the Wikipedia corpus

Deep	Source term (jap.)	Number of candidates (eng.)	Ideal translation (eng.)	Rank
0	細菌	17	Bacteria	2
1	感染	36	Infection	2
	ドイツ	4	Germany	1
3	ワクチン	88	Vaccine	2
	ヒト	472	Human	1
	微生物	84	Microorganism	1

Table 5. Example of the extracted bilingual terminology in English for the term « *santé* » in French (*deep₃*)

Source term (Fr.)	Translation candidate (Eng.)	Cosinus	Jaccard distance	Dice coefficient
Santé	creation	1,4215	0,0192	0,0377
Santé	foundation	1,4939	0,0374	0,0720
santé	preventive	1,4939	0,0374	0,0720
santé	staff	1,5018	0,0327	0,0634
santé	health	1,5024	0,065	0,1220
santé	medicine	1,5063	0,0641	0,1204
santé	confusion	1,5063	0,00944	0,0186
santé	component	1,5128	0,0248	0,0483
santé	charge	1,5135	0,0086	0,0170

Table 5. (continued)

santé	treatment	1,5221	0,0602	0,1136
santé	hospital	1,5463	0,0449	0,0860
santé	spécialisé	1,5475	0,0194	0,0380
santé	epidemiology	1,5502	0,0265	0,0517
santé	patient	1,5506	0,0335	0,0648
santé	equipment	1,5516	0,018	0,0353
santé	risk	1,5521	0,0354	0,0683
santé	approach	1,5521	0,036	0,0695

5 Conclusion

In this paper, we investigated the approach of extracting bilingual terminology from Wikipedia documents as comparable corpora in order to enrich and/or construct multilingual anthologies with the extracted terminology. We proposed a simple and adaptable approach to any language and showed preliminary evaluations for three pairs of languages including French, Japanese and English. This proposed approach showed promising results for this first study.

Among the drawbacks of the proposed approach is the introduction of many noisy terms or wrongly translations; however, most of those terms could be considered as efficient for the definition of semantic relationships in order to enrich an ontology in bilingual or multilingual format.

Further extensions include more evaluations to determine the precision and quality of translation as well as the performance of the whole system. Also, we are interested by the decomposition of the constructed large corpora using Wikipedia documents into comparable pieces or paraphrases, instead of taking the whole corpus as a single piece. Last, our main objective is the construction of a multilingual ontology and a study of several languages including those with complex morphology, such as Arabic.

References

1. Adafre, S.F., De Rijke, M.: Finding similar sentences across multiple languages in Wikipedia. In: Proceedings of the EACL Workshop on NEW TEXT Wikis and Blogs and Other Dynamic Text Sources (2006)
2. Adar, E., Skinner, M., Weld, D.S.: Information arbitrage across multi-lingual Wikipedia. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, Spain, February 09-12 (2009)
3. Dejean, H., Gaussier, E., Sadat, F.: An Approach based on Multilingual Thesauri and Model Combination for Bilingual Lexicon Extraction. In: Proceedings of COLING 2002, Taiwan (2002)
4. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics* 19(1), 61–74 (1993)

5. Erdmann, M., Nakayama, K., Hara, T., Nishio, S.: An Approach for Extracting Bilingual Terminology from Wikipedia. In: Haritsa, J.R., Kotagiri, R., Pudi, V. (eds.) DASFAA 2008. LNCS, vol. 4947, pp. 380–392. Springer, Heidelberg (2008a)
6. Erdmann, M., Nakayama, K., Hara, T., Nishio, S.: Extraction of bilingual terminology from a multilingual Web-based encyclopedia. *J. Inform. Process.* (2008b)
7. Erdmann, M., Nakayama, K., Hara, T., Nishio, S.: Improving the extraction of bilingual terminology from Wikipedia. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 5(4) (October 2009)
8. Fung, P.: A Statistical View of Bilingual Lexicon Extraction: From Parallel Corpora to Non-Parallel Corpora. In: Véronis, J. (ed.) *Parallel Text Processing* (2000)
9. Gœuriot, L., Daille, B., Morin, E.: Compilation of specialized comparable corpus in French and Japanese. *Proceedings. In: ACL-IJCNLP Workshop “Building and Using Comparable Corpora” (BUCC 2009), Singapore (August 2009)*
10. Gœuriot, L., Morin, E., Daille, B.: Reconnaissance de critères de comparabilité dans un corpus multilingue spécialisé. *Actes. In: Sixième édition de la Conférence en Recherche d’Information et Applications, CORIA 2009 (2009)*
11. Kun, Y., Tsujii, J.: Bilingual Dictionary Extraction from Wikipedia (2009a). In: *Proceedings of MT Summit XII Proceedings 2009 (2009)*
12. Kun, Y., Junichi, T.: Extracting Bilingual Dictionary from Comparable Corpora with Dependency Heterogeneity. In: *Proceedings of NAACL HLT 2009: Short Papers, Boulder, Colorado, pp. 121–124 (June 2009b)*
13. Mohammadi, M., QasemAgharee, N.: In: *Proceedings of NIPS Workshop, Grammar Induction, Representation of Language and Language Learning, Whistler, Canada (December 2009)*
14. Morin, E., Daille, B.: Extraction de terminologies bilingues à partir de corpus comparables d’un domaine spécialisé. *Traitement Automatique des Langues (TAL), Lavoisier* 45(3), 103–122 (2004)
15. Morin, E., Daille, B.: Comparabilité de corpus et fouille terminologique multilingue. *Traitement Automatique des Langues (TAL)* 47(1), 113–136 (2006)
16. Nakagawa, H.: Disambiguation of Lexical Translations Based on Bilingual Comparable Corpora. In: *Proceedings of LREC 2000, Workshop of Terminology Resources and Computation, WTRC 2000, pp. 33–38 (2000)*
17. Peters, C., Picchi, E.: Capturing the Comparable: A System for Querying Comparable Text Corpora. In: *Proceedings of the Third International Conference on Statistical Analysis of Textual Data, pp. 255–262 (1995)*
18. Rapp, R.: Automatic Identification of Word Translations from Unrelated English and German Corpora. In: *Proceedings of European Chapter of the Association for Computational Linguistics, EACL (1999)*
19. Sadat, F., Yoshikawa, M., Uemura, S.: Learning bilingual translations from comparable corpora to cross-language information retrieval: hybrid statistics-based and linguistics-based approach. In: *Proceedings of EACL 2003, Workshop on Information Retrieval with Asian Languages, Sapporo, Japan, vol. 11, pp. 57–64 (2003)*
20. Sadat, F.: Knowledge Acquisition from Collections of News Articles to Cross-language Information Retrieval. In: *Proceedings of RIAO 2004 Conference, Avignon, France, pp. 504–513 (2004)*
21. Véronis, J.: *Parallel Text Processing: Alignment and Use of Translation Corpora*. Kluwer Academic Publishers Ed., Dordrecht (2000)
22. Voss, J.: Measuring Wikipedia. In: *Proceedings of 10th International Conference of the International Society for Scientometrics and Informetrics (2005)*

Segmenting Long Sentence Pairs to Improve Word Alignment in English-Hindi Parallel Corpora

Jyoti Srivastava and Sudip Sanyal

Indian Institute of Information Technology,
Allahabad, India
{rs76,ssanyal}@iitaa.ac.in

Abstract. This paper presents an approach which improves the performance of word alignment for English-Hindi language pair. Longer sentences in the corpus create severe problems like the high computational requirements and poor quality of resulting word alignment. Here, we present a method to solve these problems by breaking the longer sentence pairs into shorter ones. Our approach first breaks the source and target sentences into clauses and then treats the resulting clause pairs as sentence pairs to train word alignment model. We also report preliminary work on automatically identifying clause boundaries which are appropriate for improvement of word alignment. This paper demonstrates the increase of precision, recall and F-measure by approximately 11%, 7%, 10% respectively and reduction in Alignment Error Rate (AER) by approximately 10% in the performance of IBM Model 1 for word alignment. These results are obtained by training on 270 sentence pair and testing on 30 sentence pairs. Experiments of this paper are based on TDIL corpus.

Keywords: Clause Identification, Word alignment, Statistical Machine Translation.

1 Introduction

Word alignment is the task of recognizing the correct translation relationships among the words of a parallel corpus [1] [2]. Word alignment is the first step of Statistical Machine Translation (SMT) and its output is also known as the translation model. This paper focuses on the word alignment of English-Hindi parallel corpus. Many word alignment techniques have been developed in Natural Language Processing (NLP) so far. Longer sentence pairs in the corpus are computationally expensive at the time of training and also degrade the performance of word alignment. We can solve this problem by breaking longer sentences into shorter one.

An effective approach to break the sentences using cue phrases is described in this paper. The basic hypothesis of this work is that if we break the longer sentences into clauses prior to the calculation of word alignment probabilities then the performance of word alignment will improve. Here word alignment

model is applied on the clause-pairs which are created from original sentence pairs of the corpus. The word alignment model used here is the IBM Model 1.

IBM Model 1 is a word alignment model which is widely used for working with parallel bilingual corpus [1]. IBM Model 1 was originally developed for providing reasonable parameter estimates to initialize more complex word-alignment models. There are also other word alignment models like IBM Models 2-5 and HMM. IBM Model 1 has the problem of fertility and distortion which is solved in IBM Models 3-5. All the IBM models are relevant because training using expectation maximization starts with the simplest IBM Model 1 for a few iterations and then proceeds through iterations of the more complex higher IBM models. So we can expect that if the output of IBM Model 1 improves, then the higher IBM Models will also get improved. Thus, we will work primarily with IBM Models 1 and expect that any improvement in this model will also get reflected in the higher models. One well known method to improve the accuracy of word alignment model is to enhance the size of the parallel corpus. But this is a very expensive process and as will be shown later, we can achieve better accuracy with our proposed method using a significantly smaller corpus. So, this paper is also an effort to deal with word alignment when resources are scarce.

2 Related Work

Words may be poorly aligned if the sentences are too long because it degrades the performance of word alignment. On the other hand, training a system using long sentence pairs is computationally expensive. In order to make good use of the information carried by long sentence pairs, it is necessary to segment long sentences into shorter ones. Xu et al. took the first step in this direction by introducing a method of doing sentence segmentation based on modified IBM Translation Model 1 [3]. To our knowledge, little further research has been done in this area. By splitting the long sentence pairs into shorter ones in the training corpus, we can get better alignment quality [3][4].

In Systran, as described by Hutchins and Somers, conjunct and relative clauses were segmented in a preprocessing step [5]. Similar methods were used in the Stanford Machine Translation project [6]. Chandrasekar applies a sentence simplification method to machine translation, where sentences are split at conjunctions, relative pronouns, etc., before translation [7]. Rao et al. describe a clause-wise translation strategy within an English- Hindi transfer-based MT system [8]. In the context of SMT, Koehn and Knight use a dedicated noun phrase (NP) translation subsystem to obtain significant improvements in German-English translation [9]. Kim and Ehara proposed a rule-based method to split long Japanese sentences to perform Japanese-to- English translation [10]. Marcu provides cue phrases for English in his thesis which can be used to break the sentences into clauses [11].

Sudoh et al. proposes a method to perform clause level alignment of the parallel corpus and to translate clauses (all clauses identified by a syntactic parser) as a unit to improve long-distance reordering in a specialized domain – English

to Japanese translation of abstracts of the research paper in the medical domain [12]. They applied automatic clause identification in both training and testing time but only at source side not on the target side. They segment the sentences into clauses at target side by using source clauses and word alignment from source language to target language. Ramanathan et al. used clause based constraints at the time of testing to improve the performance of SMT. this paper used clause identification at both side source and target but only in testing not at the time of training [13]. While our approach draws from many of the above, it is novel in the following ways:

- We are breaking source language sentences and target language sentences into clauses before training of the word alignment model.
- We used a very simple method to break the sentences into clauses by using some of the cue phrases given by Marcu and we also added some new cue phrases in this list which are used as conjunction in longer sentences [11]. These cue phrases are used as clause markers to break the sentences into clauses.
- We have created a list of cue phrases for Hindi that can be used as clause markers for target sentences in the training set.
- We demonstrate significant improvements using this strategy for word alignment in English-Hindi.

The proposed method is not restricted to a specific clause identification method. We can employ any method which satisfies the clause definition of the proposed method that clauses are independently translated. Our approach is related to sentence simplification and its intention is to obtain short and simple source and target sentences for better word alignment.

3 Proposed Approach

Problematic long sentences often include more than one clause. In such kind of sentences usually clause can be translated almost independently of words outside the clause. Longer sentences consume more time in training of word alignment and also degrade the performance. An example sentence pair from TDIL corpus is given below to demonstrate the benefit of using clause pairs instead of sentence pair for word alignment:

Sentence pair: 51(English Sentence length) * 58(Hindi Sentence length) = 2958

English: However, the focus on tourism is, indeed, gradual, and as long as any visitor is aware of its impact upon the landscape and does his utmost to limit that impact, Antarctica is still a magical experience that most, if tourism is given the chance further more, would find hard to resist. – 51

Hindi: हालांकि, पर्यटन पर केन्द्रण वास्तव में धीमा है और जब तक कोई यात्री इसके भूमि चित्र पर पड़ने वाले प्रभाव के बारे में सजग है, और उस प्रभाव को सीमित

रखने के लिये अत्यधिक प्रयासरत है, अन्टार्कटिका अभी भी एक जादुई अनुभव है यदि इस पर और अधिक पर्यटन को बढ़ावा दिया गया तो इसका प्रतिरोध मुश्किल होगा – 58

Here English sentence has 51 words and Hindi sentence has 58 words. When we will apply word alignment algorithm IBM Model 1 on this sentence pair then the algorithm will calculate probability of each of the 51 English word with each of the 58 Hindi word. So there will be total 2958 (51*58) combination of source and target word for which probability will be calculated in one iteration. And there are more than one iteration calculated to maximize the probability to get correct word alignment. so the number of computation is large. But when we break this sentence pair into clause pair as given below:

Clause pair 1: $8*8 = 64$

However, the focus on tourism is, indeed, gradual, – 8
हालांकि, पर्यटन पर केन्द्रण वास्तव में धीमा है – 8

Clause pair 2: $14*17 = 238$

and as long as any visitor is aware of its impact upon the landscape – 14
और जब तक कोई यात्री इसके भूमि चित्र पर पड़ने वाले प्रभाव के बारे में सजग है – 17

Clause pair 3: $8*11 = 88$

and does his utmost to limit that impact – 8
,और उस प्रभाव को सीमित रखने के लिये अत्यधिक प्रयासरत है – 11

Clause pair 4: $8*7 = 56$

,Antarctica is still a magical experience that most – 8
,अन्टार्कटिका अभी भी एक जादुई अनुभव है – 7

Clause pair 5: $8*10 = 80$

,if tourism is given the chance further more – 8
यदि इस पर और अधिक पर्यटन को बढ़ावा दिया गया – 10

Clause pair 6: $5*5 = 25$

,would find hard to resist. – 5
तो इसका प्रतिरोध मुश्किल होगा – 5

There are 6 clause pair in this sentence pair, the cue phrases which are used to split the sentence are shown in bold and italic in the sentence. Clause length is given in front of each clause and the total number of computation needed to calculate the word alignment probability is given in front of each clause pair. Thus the number of computation for probability calculation will be reduced upto 551 (64+238+88+56+80+25) which is very small in comparison to 2958. When number of target words are reduced to calculate the probability with each source word then probability also get improved. For example when we calculate the

probability on plain corpus then the probability of word “tourism” with “पर्यटन” is 0.81 and probability of word “impact” with “प्रभाव” is 0.06. While when the probability is calculated on the clause-separated corpus then the probability of word “tourism” with “पर्यटन” is 0.98 and probability of word “impact” with “प्रभाव” is 0.74. So when we use clause separated corpus instead of the plain corpus for word alignment the computation time is reduced as well as the performance gets improved. From this point of view, we propose an approach to use the clauses separately for word alignment. The proposed method consists of the following steps:

1. Clause Segmentation of Source Sentences
2. Training for word alignment model using these clauses

3.1 Automatic Clause Segmentation of Source and Target Sentences

A clause is a part of a sentence which contains a Subject and a Predicate. The Predicate can modify the subject. It includes a verb, objects or phrases that depend on the corresponding verb. There may be one or more than one clause in a sentence. To break the sentences into clauses, we made a list of the cue phrases for English and Hindi. For English, we used some of the cue phrases from the thesis of Marcu. We also added some cue phrases in this list to make this list complete to break the sentences into clauses. For Hindi, we created a list of cue phrases which are corresponding to the list of cue phrases for English. Now by using these cue phrases we break a sentence into clauses with Algorithm [11](#).

3.2 Training for Word Alignment Model Using These Clauses

It is possible that a source and the corresponding target sentence do not have equal number of clauses. To deal such types of case, after getting clauses, we check whether source and target sentences have the same number of clauses or not. If number of clauses is same in source and target sentence then we treat each clause pair (source side clause and corresponding target side clause) as a separate sentence pair for training otherwise we will use the complete sentences for training and not the clauses of the sentences. Now IBM Model 1 is used for word alignment on this clause-separated corpus.

4 Data and Evaluation

The system was trained on 270 sentences and tested on 30 sentences of TDIL corpus (English-Hindi). System is evaluated in terms of precision, recall and F-measure which were also frequently used in the previous word alignment literature. Och and Ney defined a fourth measure which is alignment error rate (AER) [\[14\]](#). AER is a measure of quality of word alignment. Alignment A is the set of alignments produced by the alignment model under testing. With a gold standard alignment G , each such alignment set consisting of two sets A_S ,

Algorithm 1. Clause Identification Algorithm

Input: Sentence $S(T/w_1, T/w_2, \dots, T/w_n)$ $\triangleright T$ is Part of Speech tag of word
CueList $C(C_1, C_2, \dots, C_m)$

Output: Set of Clauses $CL(CL_1, CL_2, \dots, CL_p)$

- 1: **procedure** *ClauseIdentification*(S)
- 2: $p \leftarrow 0$
- 3: $CL_p \leftarrow NULL$
- 4: **for** $i \leftarrow 1, length(S)$ **do**
- 5: $found \leftarrow 1$
- 6: **Extract** T from T/w_i
- 7: **if** $T = VB$ **then** \triangleright If T is Verb then search for Cue phrase in sentence
- 8: **for** $j \leftarrow i, length(S)$ **do**
- 9: **for** $k \leftarrow 1, m$ **do** \triangleright Search for cue phrase
- 10: **if** $w_j = C_k$ **then** \triangleright if w_j match to any cue phrase
- 11: **Append** CL_p to list CL \triangleright Add Clause CL_p to list CL
- 12: $p \leftarrow p + 1$
- 13: **Append** w_i to CL_p \triangleright Append Cue phrase to the new clause
- 14: $found \leftarrow 1$
- 15: **Jump** to Step 4.
- 16: **else**
- 17: $found \leftarrow 0$
- 18: **end if**
- 19: **end for**
- 20: **if** $found = 0$ **then**
- 21: **Append** w_i to CL_p
- 22: **Delete** T/w_i from S
- 23: **end if**
- 24: **end for**
- 25: **else**
- 26: **Append** w_i to CL_p
- 27: **Delete** T/w_i from S
- 28: **end if**
- 29: **end for**
- 30: **Append** CL_p to list CL \triangleright Add Clause CL_p to list CL
- 31: **return** CL \triangleright Return Clause list CL having all the clauses of sentence S
- 32: **end procedure**

A_P and, G_S, G_P corresponding to Sure (S) and Probable (P) alignments, these performance statistics are defined as

$$(Precision)P_T = \frac{|A_T \cap G_T|}{|A_T|} \quad (1)$$

$$(Recall)R_T = \frac{|A_T \cap G_T|}{|G_T|} \quad (2)$$

$$(F - measure)F_T = \frac{2P_T R_T}{P_T + R_T} \quad (3)$$

$$AER = 1 - \frac{|A_P \cap G_S| + |A_P \cap G_P|}{|A_P| + |G_S|} \quad (4)$$

Where T is the alignment type which can be set to either S or P .

5 Result and Discussion

We used automatic clause identification to split the sentences into clauses. This automatic clause identification method is giving almost correct result. To demonstrate this point we split 90 sentences into clauses manually as well as automatically. Thus we had two different clause separated corpus. We used both these corpus for training the word alignment models and compared the results. The results are presented in Table 1.

Table 1. Comparison of IBM Model 1 on plain corpus, Automatic clause separated corpus and Manually clause separated corpus

Corpus Size	System	Precision (%)	Recall (%)	F-measure (%)	AER (%)
90	IBM Model 1 (plain corpus)	30.39	42.42	35.41	64.59
90	IBM Model 1 (Automatic clause separated corpus)	42.70	48.48	45.41	54.59
90	IBM Model 1 (Manually clause separated corpus)	43.63	49.09	46.20	53.80

Table 1 shows the result of word alignment on plain (original) corpus, automatic clause separated corpus and manually clause separated corpus for 90 sentences. Several interesting points emerge when we examine these results. The first point to note is the difference between AER on plain corpus and clause separated corpus which is approximately 10%. Thus, clause separation gives significantly better results. The second point to note is that IBM Model 1 gives AER as 54.59% on automatic clause separated corpus and 53.80% on manually clause separated corpus of 90 sentences. So the difference between AER on automatic clause separated corpus and manually clause separated corpus is only 0.7 which is actually very small. This observation demonstrates that the automatic clause separation described by Algorithm 1 is adequate for the purposes of training word alignment models. For word alignment, we need a large corpus to get the better performance but if we split the sentences into clauses manually for this large corpus then it takes large human power. Thus it is cheap and best to use automatic clause identification method to split the sentences into clauses.

As illustrated by the Table 1, when IBM Model 1 is applied on the plain corpus of 90 sentences then AER is 64.59% but when IBM Model 1 is applied on the clause-separated corpus of same size then AER is decreased by 10% and F-measure is increased by 10% approximately. We achieved significant improvements of 12% in precision, 6% in recall. We will now demonstrate the effect of automatic clause separation as the corpus size increases.

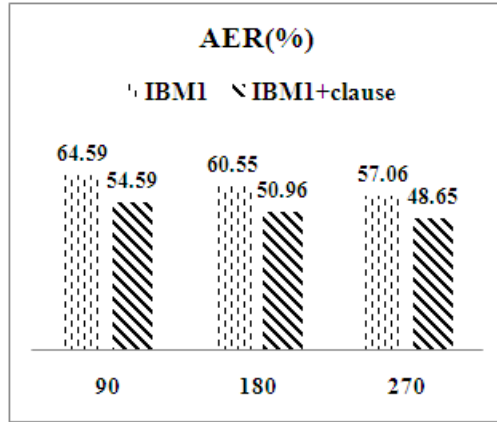


Fig. 1. Comparison of AER computed on IBM Model 1 with plain corpus and Automatically clause-separated corpus

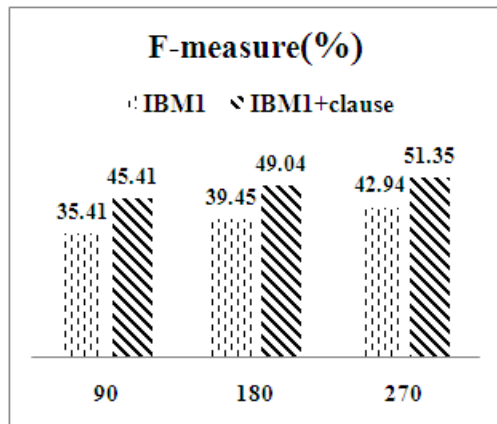


Fig. 2. Comparison of F-measure computed on IBM Model 1 with plain corpus and Automatically clause-separated corpus

We have tested our approach and the conventional IBM Models 1 for different training corpus size from 90 to 270 and the results are given in Fig. 1 and Fig. 2. IBM Model 1 algorithm which we have used, is described by Koehn in his book [15]. The result in Fig. 1 demonstrates that AER (Alignment Error Rate) decreases when the size of the parallel corpus is increased. Similarly Fig. 2 demonstrates that F-measure increases as the size of the parallel corpus is increased. This is, of course, expected and is true for any statistical process i.e. the percentage error decreases when the sample size increases. But after some limit percentage error stops to decrease. So we want to decrease this limit by

applying clause identification. The results in Fig. 1 and Fig. 2 demonstrate that by breaking the parallel corpus into clauses, we can increase the performance of the word alignment.

There are two practical benefits when we break the longer sentences into shorter ones. Training of word alignment systems tends to run faster on shorter sentences thus reduces training time. Fast training and use of short sentence in all available parallel text make it crucial for effective development of statistical NLP systems. Beyond these practical concerns, we also proved through our results that word alignments over clause pairs perform better than word alignment over sentence pairs.

F-measure can be increased and AER can be decreased by providing the solution for the following problems:

- IBM Model 1 has problems of fertility, distortion and many words to one word translation. This problem can be solved by using clause-separated corpus on higher IBM Models which solve this type of problems.
- Due to multiple Hindi equivalent words for the same English word, the frequencies of word occurrences differ significantly in the corpus and thereby jeopardize the calculations. Due to this problem, many English words are not correctly aligned.

6 Conclusion and Future Work

This paper presents a preprocessing step to improve the word alignment for English-Hindi. In this preprocessing step, we break the sentences into clauses using cue phrases. This approach contributes significantly to the reduction in Alignment Error Rate (AER). This paper focused only on clauses as segments for division. All the conducted experiments prove that using clause separated corpus with any IBM Model performs better when compared to the use of plain corpus in IBM Model 1-5, for the task of word alignment. This experiment provides new avenues to extend this approach for other Indian languages.

Although these word alignment results are encouraging, we can further improve it. In our approach we are not doing clause alignment due to which some problem occurs as explained in the example (taken from TDIL corpus) given below:

No two travelers will ever see the same icebergs forged in exactly the same form

इसका तात्कालिक सहज सौंदर्य ऐसा है

, such is its ephemeral and austere beauty

कि कोई भी दो पर्यटक समान हिमशैलों को बिल्कुल एक जैसी आकृति में ढला कभी नहीं देखते हैं

Although this is not a major problem because the word alignment approach is statistical so if this type of error is very small then it does not affect the

performance of the word alignment. By solving these problems and using higher IBM Models which also deals with fertility and distortion, we expect further improvements in the performance of word alignment.

Acknowledgments. We are thankful to IIIT Allahabad for providing the suitable infrastructure for research. This research has been funded by Tata Consultancy Services (TCS). We are really thankful to Indian Language Technology Proliferation and Deployment Centre Team for providing sample tourism parallel corpus.

References

1. Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2), 263–311 (1993)
2. Gale, W.A., Church, K.: Identifying word correspondences in parallel texts. In: Fourth DARPA Workshop on Speech and Natural Language, Asilomar, pp. 152–157 (1991)
3. Xu, J., Zens, R., Ney, H.: Sentence segmentation using IBM word alignment model 1. In: Proc. the 10th Annual Conference of the European Association for Machine Translation, Budapest, Hungary, pp. 280–287 (May 2005)
4. Meng, B., Huang, S., Dai, X., Chen, J.: Segmenting long sentence pairs for statistical machine translation. In: International Conference on Asian Language Processing, Singapore, December 7-9 (2009)
5. Hutchins, J., Somers, H.: *An Introduction to Machine Translation*, pp. 175–189. Academic Press (1992)
6. Wilks, Y.: *The Stanford Machine Translation project*, Natural Language Processing, pp. 243–290. Algorithmics Press (1973)
7. Chandrasekar, R.: *A Hybrid Approach to Machine Translation using Man Machine Communication*, Ph.D. thesis, Tata Institute of Fundamental Research, Mumbai (1994)
8. Rao, D., Mohanraj, K., Hegde, J., Mehta, V., Mahadane, P.: A practical framework for syntactic transfer of compound-complex sentences for English-Hindi machine translation. In: *Proceedings of KBCS* (2000)
9. Koehn, P., Knight, K.: Feature-rich statistical translation of noun phrases. In: *Proceedings of ACL* (2003)
10. Kim, Y.-B., Ehara, T.: A method for partitioning of long Japanese sentences with subject resolution in J/E machine translation. In: *Proc. International Conference on Computer Processing of Oriental Languages*, pp. 467–473 (1994)
11. Marcu, D.: *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*, Ph.D. thesis, Department of Computer Science, University of Toronto, Toronto, Canada (December 1997)
12. Sudoh, K., Duh, K., Tsukada, H., Hirao, T., Nagata, M.: Divide and translate: improving long distance reordering in statistical machine translation. In: *Workshop on Statistical Machine Translation and Metrics* (2010)

13. Ramanathan, A., Bhattacharyya, P., Visweswariah, K., Ladha, K., Gandhe, A.: Clause-Based Reordering Constraints to Improve Statistical Machine Translation. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, pp. 1351–1355 (November 2011)
14. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 19–51 (2003)
15. Koehn, P.: *Statistical Machine Translation*. Cambridge University Press, Published in the United States of America by Cambridge University Press, New York (2010)

Shallow Syntactic Preprocessing for Statistical Machine Translation

Hoai-Thu Vuong¹, Dao Ngoc Tu², Minh Le Nguyen³, and Vinh Van Nguyen¹

¹ Computer Science Department
University of Engineering and Technology
Vietnam National University, Hanoi
144, Xuan Thuy, Cau Giay, Hanoi
{thuvh_mcs09, vinhnv}@vnu.edu.vn

² Informatics Faculty, Hai Phong University,
171 Phan Dang Luu Street, Kien An, Hai Phong
dnthp@yahoo.com

³ Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, 923-1292, Japan
nguyenml@jaist.ac.jp

Abstract. Reordering is of essential importance for phrase based statistical machine translation. In this paper, we would like to present a new method of reordering in phrase based statistical machine translation. We inspired from [1] using preprocessing reordering approaches. We used shallow parsing and transformation rules to reorder the source sentence. The experiment results from English-Vietnamese pair showed that our approach achieves significant improvements over MOSES which is the state-of-the-art phrase based system.

Keywords: Natural Language Processing, Machine Translation, Phrase-based Statistical Machine Translation.

1 Introduction

In statistical machine translation (SMT), the reordering problem (global reordering) is one of the major problems, since different languages have different word order requirements. The statistical machine translation task can be viewed as two subtasks: predicting the collection of words in a translation, and deciding the order of the predicted words (reordering problem). Currently, phrase-based statistical machine translation [2,3] is the state-of-the-art of SMT because of its power in modelling short reordering and local context.

However, with phrase based SMT, long distance reordering is still problematic. In order to tackle the long distance reordering problem, in recent years, huge research efforts have been conducted using syntactic information. There are some studies on integrating syntactic resources within statistical machine translation. Chiang [4] shows significant improvement by keeping the strengths of phrases, while incorporating syntax into SMT. Some approaches have been

applied at the word-level [5]. They are particularly useful for language with rich morphology, for reducing data sparseness. Other kinds of syntax reordering methods require parser trees, such as the work in [6,5,7]. The parsed tree is more powerful in capturing the sentence structure. However, it is expensive to create tree structure, and building a good quality parser is also a hard task. All the above approaches require much decoding time, which is expensive.

The approach we are interested in here is to balance the quality of translation with decoding time. Reordering approaches as a preprocessing step [1,8,9,10] is very effective (improvement significant over state-of-the-art phrase-based and hierarchical machine translation systems and separately quality evaluation of reordering models).

Inspiring this preprocessing approach, we have proposed a combine approach which preserves the strength of phrase-based SMT in local reordering and decoding time as well as the strength of integrating syntax in reordering. Consequently, we use an intermediate syntax between POS tag and parse tree: shallow parsing. Firstly, we use shallow parsing for preprocessing with training and testing. Second, we apply a series of transformation rules which are learnt automatically from parallel corpus to the shallow tree. The experiment results from English-Vietnamese pair showed that our approach achieves significant improvements over MOSES which is the state-of-the-art phrase based system.

The rest of this paper is structured as follows. Section 2 reviews the related works. Section 3 briefly introduces phrase-based SMT. Section 4 introduces how to apply transformation rules to the shallow tree. Section 5 describes and discusses the experimental results. And, conclusions are given in Section 6.

2 Related Works

As mentioned in section 1, some approaches using syntactic information are applied to solve the reordering problem. One of approaches is syntactic parsing of source language and reordering rules as preprocessing steps. The main idea is transferring the source sentences to get very close target sentences in word order as possible, so EM training is much easier and word alignment quality becomes better. There are several studies to improve reordering problem such as [1,5,11,12,13,8].

They all performed reordering during preprocessing step based on the source tree parsing combining either automatic extracted syntactic rules [1,11,13] or manually written rules [5,12,8].

[8] described method using dependent parse tree and a flexible rule to perform the reordering of subject, object, etc... These rules were written by hand, but [8] showed that an automatic rule learner can be used.

[5] developed a clause detection and used some handwritten rules to reorder words in the clause. Partly, [1,13] built an automatic extracted syntactic rules.

Compared with these approaches, our work has a few differences. Firstly, we aim to develop the phrase-based translation model to translate from English to Vietnamese. Secondly, we build a shallow tree by chunking in recursively (chunk

of chunk). Thirdly, we use the automatic rules, which is learnt automatically from parallel corpus, to transform the source sentence. As the same with [11,13], we also apply preprocessing in both training and decoding time.

The other approaches use syntactic parsing to provide multiple source sentence reordering options through word (phrase) lattices [14,15]. [15] applied some transformation rules, which is learned automatically from bilingual corpus, to reorder some words in a chunk. A crucial difference between their methods and ours is that they do not perform reordering during training. While, our method can solve this problem by using a complicated structure, which is more efficient with a shallow tree (chunk of chunks).

3 Brief Description of the Baseline Phrase-Based SMT

In this section, we will describe the phrase-based SMT system which was used for the experiments. Phrase-based SMT, as described by [2] translating a source sentence into a target sentence by decomposing the source sentence into a sequence of source phrases, which can be any contiguous sequences of words (or tokens treated as words) in the source sentence. For each source phrase, a target phrase translation is selected, and the target phrases are arranged in some order to produce the target sentence. A set of possible translation candidates created in this way is scored according to a weighted linear combination of feature values, and the highest scoring translation candidate is selected as the translation of the source sentence. Symbolically,

$$\hat{t} = \arg \max_{t,a} \sum_{i=1}^n \lambda_i f_j(s, t, a) \quad (1)$$

when s is the input sentence, t is a possible output sentence, and a is a phrasal alignment that specifies how t is constructed from s , and \hat{t} is the selected output sentence. The weights λ_i associated with each feature f_i are tuned to maximize the quality of the translation hypothesis selected by the decoding procedure that computes the argmax. The log-linear model is a natural framework to integrate many features. The baseline system uses the following features:

- the probability of each source phrase in the hypothesis given the corresponding target phrase.
- the probability of each target phrase in the hypothesis given the corresponding source phrase.
- the lexical score for each target phrase given the corresponding source phrase.
- the lexical score for each source phrase given the corresponding target phrase.
- the target language model probability for the sequence of target phrase in the hypothesis.
- the word and phrase penalty score, which allow to ensure that the translation does not get too long or too short.
- the distortion model allows for reordering of the source sentence.

The probabilities of source phrase given target phrases, and target phrases given source phrases, are estimated from the bilingual corpus.

[2] used the following distortion model (reordering model), which simply penalizes nonmonotonic phrase alignment based on the word distance of successively translated source phrases with an appropriate value for the parameter α :

$$d(a_i - b_{i-1}) = \alpha^{|a_i - b_{i-1} - 1|} \quad (2)$$

4 Shallow Syntactic Preprocessing for SMT

In this section, we describe the transformation rules and how applying it to shallow tree for reordering an English sentence.

4.1 Transformation Rule

Suppose that T_s is a given lexicalized tree of the source language (whose nodes are augmented to include a word and a POS label). T_s contains n applications of lexicalized CFG rules $LHS_i \rightarrow RHS_i$ ($i \in \overline{1, n}$). We want to transform T_s into the target language word order by applying transformational rules to the CFG rules. A transformational rule is represented as $(LHS \rightarrow RHS, RS)$, which is a pair consisting of an unlexicalized CFG rule and a reordering sequence (RS). For example, the rule $(NP \rightarrow JJ NN, 1\ 0)$ implies that the CFG rule $(NP \rightarrow JJ NN)$ in the source language can be transformed into the rule $(NP \rightarrow NN JJ)$ in the target language. Since the possible transformational rule for each CFG rule is not unique, there can be many transformed trees. The problem is how to choose the best one (we can see [11] for a description in more detail). We use the method described in [11] to extract the transformation rules from the parallel corpus and induce the best sequence of transformation rules for a source tree.

4.2 Shallow Syntactic Processing

In this section, we describe a method to build a translation model for a pair English to Vietnamese. We aim to reorder an English sentence to get a new English, and some words in this sentence are arranged as Vietnamese words order.

Figure 1 gives examples of original and preprocessed phrase in English. The first line is the original English phrase with a chunk (two blue books), and the second line is the phrase with a modified chunk (two books blue). This chunk is arranged as the Vietnamese order. However, we aim to preprocess the words outside the chunk (the phrase "tom 's" in Figure 1), and the third line is the output of our method. Finally, the fourth line is the Vietnamese phrase. As you can see, the third and fourth line have the same word order.

After pre-processed, this new sentence is used in training process to get a phrased translation model, and in decoding process to get a target sentence (by using translation which is trained in training process). To pre-process, we follow these steps:

tom 's [two blue books]
 tom 's [two books blue]
 [two books blue] 's tom
 hai cuốn sách màu xanh của tom

Fig. 1. An Example of phrase before and after our preprocessing

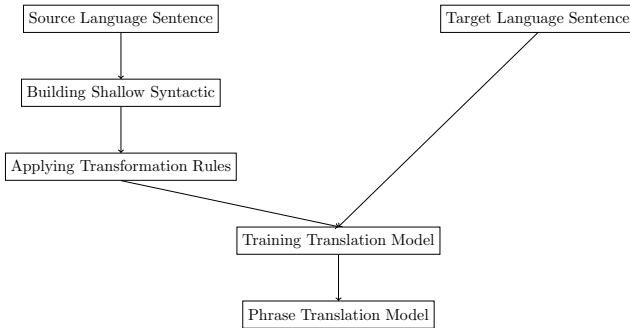


Fig. 2. Our training process

- building shallow syntactic
- applying transformation rules

So as to build shallow syntactic, we use a method described in [16]. Their approach introduced the method to parse an English sentence by using chunking (balance accuracy with speed time). Their method is high accuracy (accuracy with 88.4 F-score) and fast parsing time: using CRFTagger to chunk the sentence, and then setup a tree from the chunks and recursive until they cannot chunk the sentence. Their results showed that this method is outstanding in performance with high accuracy. As they did, we also receive a shallow syntactic when parse the source sentence in English. However, we stop chunking after two loop steps. So that, *the highest deep of node in syntactic tree is two*. By doing that, we will balance between accuracy and performance time. We can use the method of [16] to build full parse tree, but that will be leave it for future work.

After building the shallow syntactic, the transformation rules are applied. After finding the matching rule from the top of the shallow tree, we arrange the words in the English sentence, which is covered by the matching node, like Vietnamese words order. And then, we do the same for each children of this node. If any rule is applied, we use the order of original sentence. Noticed that, these rules are extracted automatically from bilingual corpus like method of Nguyen [11].

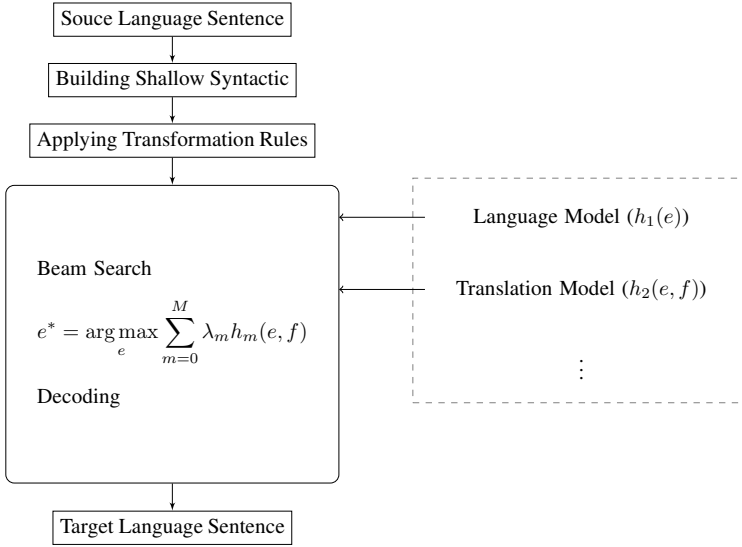


Fig. 3. Our decoding process

5 Experiment

5.1 Implementation

- We developed the shallow parsing by using the method from [16] to parse a source sentences (English sentences) including a shallow tree.
- The rules are learnt from English-Vietnamese parallel corpus and Penntree Bank Corpus. We used the CFG transformation rules (chunk levels) for extraction from [11]’s method to reorder shallow tree of a source sentences.
- We implemented preprocessing step during both training and decoding time.
- Using the SMT Moses decoder [17] for decoding.

5.2 Data Set and Experimental Setup

For evaluation, we used an English-Vietnamese corpus [18], including about 54642 pairs for training, 500 pairs for testing and 200 pairs for development test set. Table 1 gives more statistical information about our corpora. We conducted some experiments with SMT Moses Decoder [17] and SRILM [19]. We trained a trigram language model using interpolate and kndiscount smoothing with 89M Vietnamese mono corpus. Before extracting phrase table, we use GIZA++ [20] to build word alignment with grow-diag-final-and algorithm. Besides using preprocessing, we also used default reordering model in Moses Decoder: using word-based extraction (wbe), splitting type of reordering orientation to three class (monotone, swap and discontinuous – msd), combining backward and forward

Table 1. Corpus Statistical

Corpus	Sentence pairs	Training Set	Development Set	Test Set
General	55341	54642	200	499
			English	Vietnamese
Training	Sentences		54620	
	Average Length		11.2	10.6
	Word		614578	580754
	Vocabulary		23804	24097
Development	Sentences		200	
	Average Length		11.1	10.7
	Word		2221	2141
	Vocabulary		825	831
Test	Sentences		499	
	Average Length		11.2	10.5
	Word		5620	6240
	Vocabulary		1844	1851

direction (bidirectional) and modeling base on both source and target language (fe) [17]. To contrast, we try preprocessing the source sentence with some hand-written rules and automatically rules, which is described in [41]. In addition, we did experiments with a base chunk and our shallow syntactic tree. Finally, by setting a flag for Moses Decoder, we use a monotone decoder to carry out the effective of our method.

Table 2. Details of our experimental, AR is named as using automatic rules

Name	Description
Baseline	Phrase-based system
Baseline + AR	Phrase-based system with corpus which is preprocessed using automatic learning rules
Baseline + AR (monotone)	Phrase-based system with corpus which is preprocessed using automatic learning rules and decoded by monotone decoder
Baseline + AR (shallow syntactic)	Phrase-based system with corpus which is shallow syntactic analyze and applied automatic transformation rules
Baseline + AR (shallow syntactic+monotone)	Phrase-based system with corpus which is shallow syntactic analyze and applied automatic transformation rules

5.3 BLEU Score

The result of our experiments in table 3 showed our applying transformation rule to process the source sentences. Thanks to this method, we can find out various phrases in the translation model. So that, they enable us to have more options for decoder to generate the best translation.

Table 4 describes the BLEU score [21] of our experiments. As we can see, by applying preprocess in both training and decoding, the BLEU score of our

Table 3. Size of phrase tables

Name	Size of phrase-table
Baseline	1237568
Baseline + AR	1243699
Baseline + AR (monotone)	1243699
Baseline + AR (shallow syntactic)	1279344
Baseline + AR (shallow syntactic + monotone)	1279344

best system increase by 0.82 point "Baseline + AR (shallow syntactic)" system) over "Baseline system". Improvement over 0.82 BLEU point is valuable because baseline system is the strong phrase based SMT (integrating lexicalized reordering models). The improvement of "Baseline + AR (shallow syntactic)" system is statistically significant at $p < 0.01$.

Table 4. Translation performance for the English-Vietnamese task

System	BLEU (%)
Baseline	36.84
Baseline + AR	37.24
Baseline + AR (monotone)	35.80
Baseline + AR (shallow syntactic)	37.66
Baseline + AR (shallow syntactic + monotone)	37.43

Finally, the BLEU score of using monotone decoder decrease by 1% when we use preprocessing in only base chunk level, and our shallow syntactic decreased a bit. As, the default reordering model in baseline system is better than in this experiment¹.

6 Conclusion

In this paper, we would like to present a new method for reordering in phrase based statistical machine translation. We inspired from [1] using preprocessing reordering approaches. We used shallow parsing and transformation rules for reordering the source sentence. Meanwhile, we limit the height of syntactic tree to balance the accuracy with performance of system. The experiment results with English-Vietnamese pair showed that our approach achieves significant improvements over MOSES which is the state-of-the art phrase based system. In the future, we would like to evaluate our method with tree with higher and deeper syntactic structure and larger size of corpus.

¹ The reordering model in the monotone decoder is distance based, introduced in [2]. This model is a default reordering model in Moses Decoder [17].

Acknowledgment. This work described in this paper was partially supported by TRIG-B project (EEC5.3B).

References

1. Xia, F., McCord, M.: Improving a Statistical MT System with Automatically Learned Rewrite Patterns. In: Proceedings of Coling 2004, Geneva, Switzerland, August 23–August 27, pp. 508–514 (2004)
2. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of HLT-NAACL 2003, Edmonton, Canada, pp. 127–133 (2003)
3. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. *Computational Linguistics* 30(4), 417–449 (2004)
4. Chiang, D.: A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), Ann Arbor, Michigan, pp. 263–270 (June 2005)
5. Collins, M., Koehn, P., Kucerová, I.: Clause restructuring for statistical machine translation. In: Proc. ACL 2005, Ann Arbor, USA, pp. 531–540 (2005)
6. Quirk, C., Menezes, A., Cherry, C.: Dependency treelet translation: Syntactically informed phrasal smt. In: Proceedings of ACL 2005, Ann Arbor, Michigan, USA, pp. 271–279 (2005)
7. Huang, L., Mi, H.: Efficient incremental decoding for tree-to-string translation. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 273–283. Association for Computational Linguistics, Cambridge (2010)
8. Xu, P., Kang, J., Ringgaard, M., Och, F.: Using a dependency parser to improve smt for subject-object-verb languages. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 245–253. Association for Computational Linguistics, Boulder (2009)
9. Talbot, D., Kazawa, H., Ichikawa, H., Katz-Brown, J., Seno, M., Och, F.: A lightweight evaluation framework for machine translation reordering. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, pp. 12–21. Association for Computational Linguistics, Edinburgh (2011)
10. Katz-Brown, J., Petrov, S., McDonald, R., Och, F., Talbot, D., Ichikawa, H., Seno, M., Kazawa, H.: Training a parser for machine translation reordering. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, pp. 183–192. Association for Computational Linguistics, Scotland (2011)
11. Nguyen, T.P., Shimazu, A.: Improving phrase-based smt with morpho-syntactic analysis and transformation. In: Proceedings AMTA 2006 (2006)
12. Wang, C., Collins, M., Koehn, P.: Chinese syntactic reordering for statistical machine translation. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 737–745. Association for Computational Linguistics, Prague (2007)
13. Habash, N.: Syntactic preprocessing for statistical machine translation. Proceedings of the 11th MT Summit (2007)

14. Zhang, Y., Zens, R., Ney, H.: Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In: Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation, pp. 1–8 (2007)
15. Nguyen, P.T., Shimazu, A., Nguyen, L.-M., Nguyen, V.-V.: A syntactic transformation model for statistical machine translation. *International Journal of Computer Processing of Oriental Languages (IJCPOL)* 20(2), 1–20 (2007)
16. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: Fast full parsing by linear-chain conditional random fields. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2009, pp. 790–798. Association for Computational Linguistics, Stroudsburg (2009)
17. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of ACL, Demonstration Session (2007)
18. Nguyen, T.P., Shimazu, A., Ho, T.B., Nguyen, M.L., Nguyen, V.V.: A tree-to-string phrase-based model for statistical machine translation. In: Proceedings of the Twelfth Conference on Computational Natural Language Learning (CoNLL 2008), Coling 2008 Organizing Committee, pp. 143–150 (August 2008)
19. Stolcke, A.: Srlm - an extensible language modeling toolkit. In: Proceedings of International Conference on Spoken Language Processing, vol. 29, pp. 901–904 (2002)
20. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1), 19–51 (2003)
21. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, pp. 311–318 (July 2002)

Linguistic Rules Based Approach for Automatic Restoration of Accents on French Texts

Paul Brilliant Feuto Njonko, Sylviane Cardey-Greenfield, and Peter Greenfield

Centre Tesnière - Équipe d'Accueil EA 2283
Université de Franche-Comté - UFR SLHS
30, rue Mégevand, 25030 Besançon Cedex, France
paul.feuto_njonko@edu.univ-fcomte.fr,
{sylviane.cardey,peter.greenfield}@univ-fcomte.fr
<http://tesniere.univ-fcomte.fr>

Abstract. Nowadays, in the context of email as well as many other domains, there are more and more French texts wrongly accented or completely unaccented. Furthermore, it should be noted that in French, the accent has a value and a linguistic function. It expresses the languages subtleties and especially allows avoiding ambiguities and misinterpretation. Even though in most cases the loss of information resulting from the absence of accents is not a major issue for human beings, it is very problematic for automatic processing of text and increases the ambiguity involved in Natural Language Processing. However, it gets tedious to do this manually hence the importance of automatic accent restoration systems. In this perspective, this paper aims at presenting a novel system for the automatic restoration of accents in French texts. Unlike a few existing approaches using statistical methods, our approach is essentially based on linguistic rules that are more reliable.

Keywords: Natural Language Processing, Automatic Restoration of Accents, Linguistic Rules.

1 Introduction

In many Natural Language Processing (NLP) applications addressing the automatic processing of texts (Information Extraction, Machine Translation, etc.), the outcomes obtained depend on the quality of the input texts. The more these texts are noisy, the less the results are relevant, the more they are pre-processed, the more relevant the results are. In French texts, the inappropriate use or the absence of accents is becoming more widespread especially in the context of email and thereby increases the degree of ambiguity thus subverting the automatic processing of such texts. It should be noted that in French the accent is very useful and cannot be simply ignored. It expresses the languages subtleties and especially allows avoiding ambiguities and misinterpretation. For instance:

*Il dort **ou** il travaille // Il dort **où** il travaille.*
(He sleeps **or** he works // He sleeps **where** he works).

Thus, this problem should be addressed for such texts before any automatic processing in order to achieve better results in subsequent analyses. Even though in most cases the loss of information resulting from the absence of accents is not a major issue for human beings, it is very problematic for automatic text processing because information is crucial and needs to be processed with as few ambiguities as possible. All these points argue for accented source texts in French. However, it gets tedious to do this manually for reasons such as time, effort, reliability etc. In this perspective, this paper aims at presenting a novel system for automatic restoration of accents on French texts. Unlike the few existing approaches using statistical methods, our approach is essentially based on linguistic rules that are more reliable. To this end, we first established a formal linguistic model and implemented it afterwards to produce a tool for the automatic restoration of accents.

2 Related Work

In the literature, previous work has addressed the field of automatic restoration of accents of unaccented French texts, resulting in automatic tools. The most cited are described in [1], [2] and [3]. All use a probabilistic approach and their process consists in:

- **Segmentation** which consists in identifying the unaccented word on which the treatment will be performed.
- **Hypothesis generation** which consists in producing the list of all accentuation possibilities for each word identified in the segmentation process. For example, if the unit *cote* has been isolated, the system would have to generate *cote*, *coté*, *côte*, and *côté*.
- **Disambiguation** which consists in choosing the most probable accentuation. To do this, a stochastic language model called Hidden Markov Model (HMM) is used.

In contrast, our approach is essentially based on linguistic rules that are more reliable than probabilistic approaches. We do not use large lexicons for accentuation, and this allows better processing of unknown words. The processing unit remains the word, but we have studied the different rules of usage of the most used accents on the vowels in French: acute accent, grave accent, and circumflex accent. For instance, for the vowel 'e', we have analyzed its different positions in the word where it might take an acute accent, grave accent or circumflex accent.

3 Description of the System

3.1 Linguistic Analysis

Since our approach is essentially rule-based, we carried out a linguistic analysis of studied accents (acute accent, grave accent, and circumflex accent) their different rules of usage on the vowels in French drawn from reference grammars and

spelling guides [4], [5], [6]. However, we faced many problems because most rules described in grammar books were based on phonetic criteria. For instance:

- Acute accent is used in closed 'e' (*e fermé*), [5]
- Grave accent is used in open 'e' (*e ouvert*), [5]

Although these rules presented in many grammar and spelling books provide invaluable information on acute and grave accents, they remain however insufficient for their automatic processing. Phonetic rules are difficult to implement and our system needs rules based on graphic criteria. Thus, we extended our research most on the automatic processing of accents [7], [8], [9]. We came up with consistent linguistic rules based on graphic criteria governing the usage of accents in all positions in words (initial, middle and final). We then produced a linguistic formalization in order to implement our automatic accent restoration system.

3.2 General Architecture of the System

In this section, we present the general architecture of the system in terms of its components as illustrated in Fig. 1. The system takes as input an unaccented text and applies a set of accentuation rules. If the original text contains accents, then the user can choose firstly to have these be removed by the system

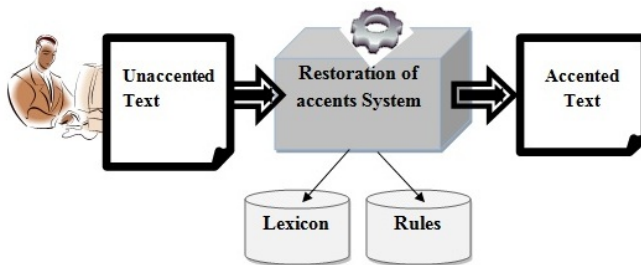


Fig. 1. General architecture of our system

Lexicon. The lexicon is our database which contains all the lists of words accessed by the system during the accentuation process. These lists are constituted of exceptions to the different rules, etc. Thus, whilst the system accesses the lexicon, only the appropriate lists for the rules under consideration are used.

Rules. This is the database which contains all the accentuation rules. It is also divided into sub-rules specific to each of the accents (acute, grave, and circumflex).

3.3 Detailed Architecture of the System

The detailed architecture of our system shown in Fig. 2 presents the route followed by a word to be accented. The system takes as input an unaccented word and performs a surface lexical analysis in order to apply different rules. Our analysis is limited in terms of the processing of homographs and compound words, which would require syntactic and semantic analysis for their recognition.

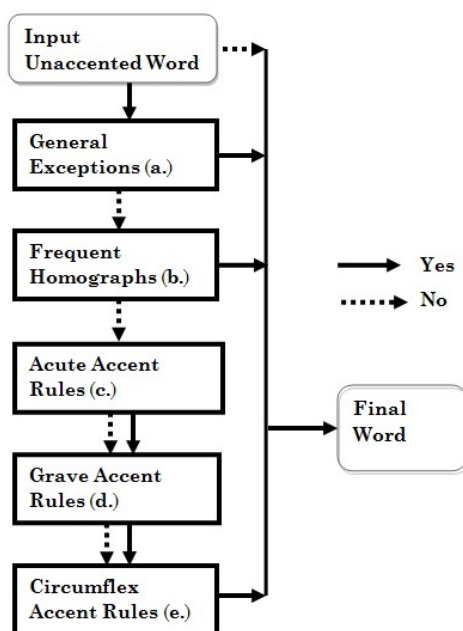


Fig. 2. Detailed architecture of our system

The processing is performed as follows:

- a.** Does the word belong to the list of general exceptions? If yes, it remains unaccented and this is the end of process. Otherwise, go to **b.**
- b.** Is the word a frequent homograph? If yes, it remains unaccented and this is the end of process. Otherwise, go to **c.**
- c.** The word enters the verification sub-module of acute accent rules. The system starts by checking whether it belongs to acute accent exceptions, if yes, it goes directly to **d.** Otherwise, it applies to the word the acute accent rules starting from the less general to the most general before going to **d.**

d. The word enters the verification sub-module of grave accent rules. The system starts by checking whether it belongs to grave accent exceptions, if yes, it goes directly to **e.** Otherwise, it applies to the word the grave accent rules starting from the less general to the most general before going to **e.**

e. The word enters the verification sub-module of circumflex accent rules. The system starts by checking whether it belongs to circumflex accent exceptions, if yes, end of process. Otherwise, it applies to the word the circumflex accent rules starting from the less general to the most general before going to the end of process.

4 Formalization of Accentuation Rules

We present here the micro-systemic algorithmic representation [10] which consists in representing rules in a structured way in order to encode them in a programming language. With this representation, they can be used not only by our system, but also by other applications.

4.1 Operators

Operators or treatments are the operations performed when a rule is true or not. For our application, these are as follows:

- A : 'e' takes an acute accent
- $B1$: 'e' takes a grave accent
- $B2$: 'a' takes a grave accent
- $C1$: 'e' takes a circumflex accent
- $C2$: 'a' takes a circumflex accent
- $C3$: 'i' takes a circumflex accent
- $C4$: 'u' takes a circumflex accent
- R : the word is a frequent homograph

The negation operator is the same operator on which we have added the logic operator \neg ($\neg A$, $\neg B$, $\neg C1$, etc.)

4.2 Formal Notation of a Rule

A formal rule is a relationship between a condition and an operator. If the condition is true, then the system performs the associated operator. We have attached to each operator a parameter n indicating more information about the rule

$$\textit{Condition} \implies \textit{Operator}(n)$$

$n = 0$: the rule does not have exceptions

$n = \textit{lists}$: the rule has a list of exceptions

When a general rule has a list of exceptions, these exceptions are going to be associated with an operator which could be the negation of the one applied to the general rule or any operator else. It should be noted that during execution, exceptions are first executed before the general rule. For instance:

$$e + \textit{consonant} + \textit{vowel} \implies A(\textit{lists})$$

Thus, with rules similar to the one above, we have formally represented the set of rules (about one hundred and fifty) of accentuation which are implemented in our automatic tool.

5 Implementation and Evaluation

5.1 Implementation

Our formal linguistic model made up of all the set of rules has been implemented using Python to produce an automatic tool for accentuation. Fig. 3 below presents the graphical user interface of the system. The user can either enter their text directly in the editor or open an existing file for accentuation.

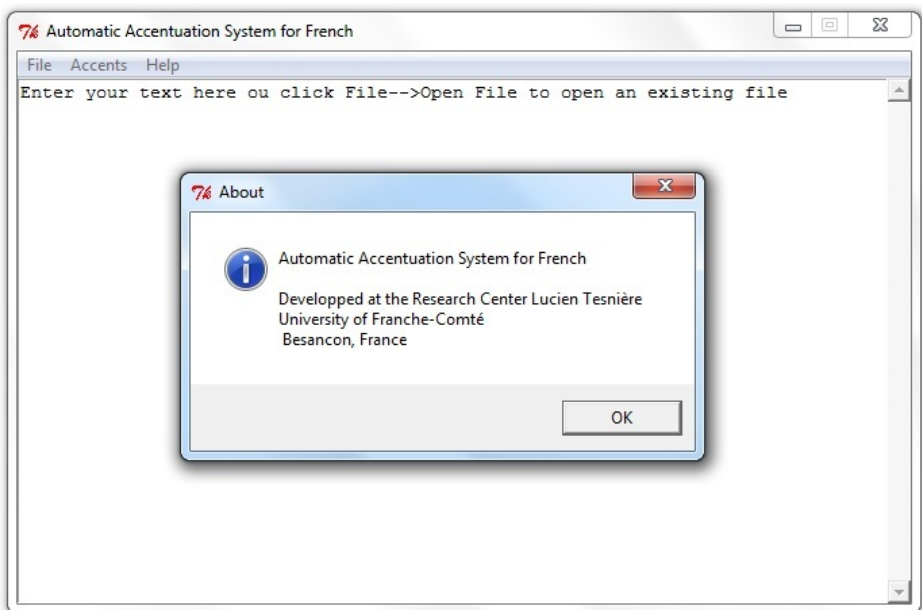


Fig. 3. User interface of the system

Fig. 4 shows an example of a sentence without accents input to the system and Fig. 5 shows the restoration of accents made by the system.

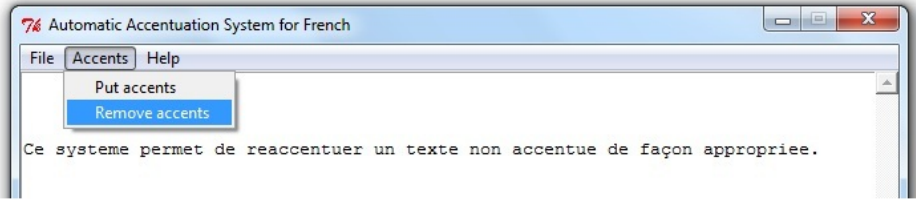


Fig. 4. An example of an unaccented sentence input to the system

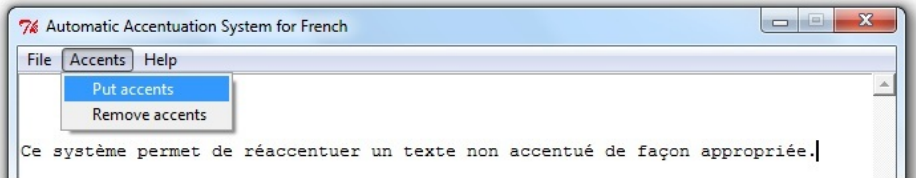


Fig. 5. Example of accentuation made by the system

5.2 Evaluation

We have evaluated our system on different texts (emails, surveys, etc.). The results shown in Table 1 present the evaluation done on corpus of about 10,000 words.

Table 1. Evaluation of the system

Number of words	Ambiguities (end in à /a)	Ambiguities (end in é /e)	Others (lo-cutions, etc.)	Non ambiguous	Total
Total	1500	750	250	7,500	10,000
Correct	1350	450	215	6,500	8,515
Accuracy	90%	60%	86%	86.6%	85.5%
Error rate	10%	40%	14%	13.4%	14.5%

6 Conclusion and Future Work

We have presented in this paper a novel system for automatic restoration of accents in French texts. In reality, the absence of accents increases the inherent ambiguity involved in such texts and could prevent their automatic processing.

This concern has prompted previous research by others that has led to the construction of tools for automatic restoration of accents. In contrast with these approaches which are statistically based, the system presented in this paper follows a completely different approach and is essentially rule based. This approach is more reliable and does not use a very large lexicon; furthermore it allows better processing of unknown words. We evaluated our system and obtain 85.5 % accuracy.

However, a number of limitations need to be considered. Currently, the system faces some problems related to the processing of complex homographs and compound words because the treatment is limited to the surface analysis of words. This additional processing requires a syntactic and/or semantic study of the context around the words. Our future work is focused on syntactic and semantic analysis that will help to overcoming these limitations in order to achieve better performance.

References

1. Simard, M.: Réaccentuation automatique de textes français. Centre d'innovation en technologies de l'information (CITI), Laval (1996)
2. El-beze, M., Spriet, T.: Réaccentuation automatique de textes. Laboratoire Informatique d'Avignon, LIA (1996)
3. Mary, V., Le beux, P.: Grepator: Accents & Case Mix for Thesaurus. In: Connecting Medical Informatics and Bio-Informatics: Proceedings of the XIXth International Congress of the European Federation for Medical Informatics, pp. 787–792. IOS Press (2005)
4. Imprimerie, N.: Lexique des règles typographiques en usage à l'Imprimerie nationale. Imprimerie nationale (2002)
5. Grevisse, M., Goosse, A.: le bon usage électronique: grammaire française, 14th edn., de boeck duculot (2007)
6. Doppagne, A.: Majuscules, abréviations, symboles et sigle pour une toilette parfaite du texte, 3e édition, Paris, Bruxelles, Duculot (1998)
7. Bioud, M.: Une normalisation sur l'emploi de la majuscule et sa représentation formelle pour un système de vérification automatique des majuscules dans un texte: thèse de doctorat, Centre de recherche Lucien Tesnière, Université de Franche-Comté (2006)
8. Al-Shafi, B.: Traitement informatique des signes diacritiques, pour une application automatique et didactique: thèse de doctorat, Centre de recherche Lucien Tesnière, Université de Franche-Comté (1996)
9. Feuto, N.P.B.: Rule based approach for normalizing messages in the security domain. In: Natural Language Processing and Human Language Technology, BULAG n36, PUFC (2011) ISSN 0758 6787
10. Cardey, S., Greenfield, P.: A Core Model of Systemic Linguistic Analysis. In: Proceedings of the International Conference RANLP 2005 Recent Advances in Natural Language Processing, Borovets, Bulgaria (September 2005)

Word Clustering for Persian Statistical Parsing

Masood Ghayoomi

German Grammar Group, Freie Universität Berlin, Germany
masood.ghayoomi@fu-berlin.de

Abstract. Syntactically annotated data like a treebank are used for training the statistical parsers. One of the main aspects in developing statistical parsers is their sensitivity to the training data. Since data sparsity is the biggest challenge in data oriented analyses, parsers have a malperformance if they are trained with a small set of data, or when the genre of the training and the test data are not equal. In this paper, we propose a word-clustering approach using the Brown algorithm to overcome these problems. Using the proposed class-based model, a more coarser level of the lexicon is created compared to the words. In addition, we propose an extension to the clustering approach in which the POS tags of the words are also taken into the consideration while clustering the words. We prove that adding this information improves the performance of clustering specially for homographs. In usual word clusterings, homographs are treated equally; while the proposed extended model considers the homographs distinct and causes them to be assigned to different clusters. The experimental results show that the class-based approach outperforms the word-based parsing in general. Moreover, we show the superiority of the proposed extension of the class-based parsing to the model which only uses words for clustering.

Keywords: Statistical Parsing, Word Clustering, the Persian Language.

1 Introduction

Parsing a natural language aims to provide a syntactic analysis of a sentence. To achieve this goal automatically, a parser, either rule-based or statistical, should be used. Data oriented parsers are trained with annotated data, like a treebank. Contrary to rule-based parsers, the statistical parsers are very sensitive to the data they are trained with, and one big problem of the training data is that it is always sparse. As a result, it is very difficult to build an accurate model from sparse data. Additionally, it is very likely to face unknown words while parsing in real applications.

Word clustering has caught attention in natural language processing to represent a coarser level of the lexical information rather than the words themselves. In this approach, words are clustered in an off-line process based on their occurrence in an unannotated corpus through an unsupervised method. In our study, we aim to use a word clustering approach for parsing to improve the performance of our statistical parser for Persian trained with a very small amount of

data. One important problem of word clustering is that homographs are treated equally which leads them to be clustered inaccurately. In addition to the class-based parsing, we propose a model which uses the part-of-speech (POS) tags of the words as an important additional lexical information in clustering to distinct the homographs and to cluster them into different classes consequently.

The structure of this paper is as follows: in Section 2 we briefly describe some basic properties of Persian. In Section 3, the tool used for parsing Persian is explained. Section 4 devotes to the treebank used for training the parser. Section 5 describes class-based parsing and the Brown algorithm used for this aim. Section 6 explains the setup of the experiments for the proposed parsing models and the obtained results; and finally, the paper is summarized in Section 7.

2 The Persian Language

Persian is a member of the Indo-European language family and it has many features in common with the other languages of this family in terms of phonology, morphology, syntax, and lexicon. Persian uses a modified version of the Arabic script and it is written right-to-left. However, the two languages differ from one another in many respects. Persian belongs to the subject-drop languages with an SOV constituent order in unmarked constructions. The constituent order is relatively free. Verbs are inflected for tense and aspect, and they agree with the subject in person and number. The language does not make use of gender [19]. There exists a so called ‘pseudo-space’ in the internal structure of the Persian lexical items. Using a white space rather than ‘pseudo-space’ will intensify the multi-word token problem. Moreover, contrary to long vowels, short vowels usually are not written but they are pronounced. This property leads to have more homographs in written texts.

3 Stanford Parser for Persian

The Stanford parser is a Java implementation of a lexicalized, probabilistic natural language parser [14]. The parser is based on an optimized Probabilistic Context Free Grammar (PCFG) and lexicalized dependency parsers, and a lexicalized PCFG parser. The output of the parser provides the phrase structure tree of a sentence along with the dependencies of the words in the sentence.

Three basic modules, namely *FactoredLexicon*, *ChineseLexicon*, and *BaseLexicon* modules, are defined in the parser for learning the lexicon. The most important task of these modules is to calculate the probability of a word given its tag, $P(word|tag)$, to let the parser choose the best tag of the word in the local context, and to utilize this probability to find the best tree structure for a sentence. The *BaseLexicon* module learns the lexicon from the training data, say the treebank. This module has worked quite appropriately for Penn English Treebank. In the adaptation of the Stanford parser for Persian, we have used the *BaseLexicon* module as well.

It should be added that a morphological tokenizer and lemmatizer are required within the Stanford CoreNLP package. Since this package currently lacks these tools for Persian, the parser assumes that tokenization and lemmatization have already been done both on the training data and test data. However, these shortcomings may affect the performance of the parser in real applications.

Following the study of Collins [9], to make the parser able to work with a data from a treebank, it is required to provide the list of heads in the phrase structure trees. To define the heads semi-automatically for Persian, we extracted all the grammar rules from the Persian treebank (PerTreeBank) and based on the labels of the mother nodes, we determined the heads of the constituents for the parser.

4 The Persian Treebank

PerTreeBank¹ is the first treebank for Persian which is developed in the framework of the HPSG [23] formalism and it is freely available on-line. No feature structures are used in the development of this treebank, but basic properties of HPSG are simulated. This treebank contains 1012 trees from the Bijankhan Corpus² and it is developed semi-automatically via a bootstrapping approach [11][2]. This treebank which has the XML data structure provides the phrase structure trees of the sentences in the Chomskyan grammar such that the type of the dependencies in the nodes' relations of the mother nodes are defined explicitly according to the basic schemas in HPSG, namely head-subject, head-complement, head-adjunct, and head-filler to bind off the extraposed constituents. It needs to be added that the canonical positions of the scrambled or extraposed elements are explicitly determined with the *nid* (not immediate dominance) node; therefore trace-based analyses of sentences are provided in PerTreeBank. Moreover, elliptical elements are also determined explicitly with a node which defines the type of ellipsis. The available morpho-syntactic and semantic information of the words in the Bijankhan Corpus is also used for the words of the treebank; as a result, the treebank is rich both in terms of the available information of the POS tags and the tree analyses of the sentences. Figure 1 displays the tree representation of sample (1):

- (1) born be donbāle in ast ke čizi rā besāzad
 Born to follow.EZ this is that something.RES DOM SUBJ.create.3SG
 ke qablan vojūd nadāšteast.
 that before existence NEG.had.CL.3sg
 ‘Born is after this [namely] to create something that did not exist before.’

To use the treebank for our experiments, we need to normalize the trees and convert the treebank from the XML format into a plain text Penn Treebank style. To this end, several conversion is done on the treebank. As said, Persian is

¹ <http://hpsg.fu-berlin.de/~ghayoomi/PTB.html>

² <http://ece.ut.ac.ir/dbrg/bijankhan/>

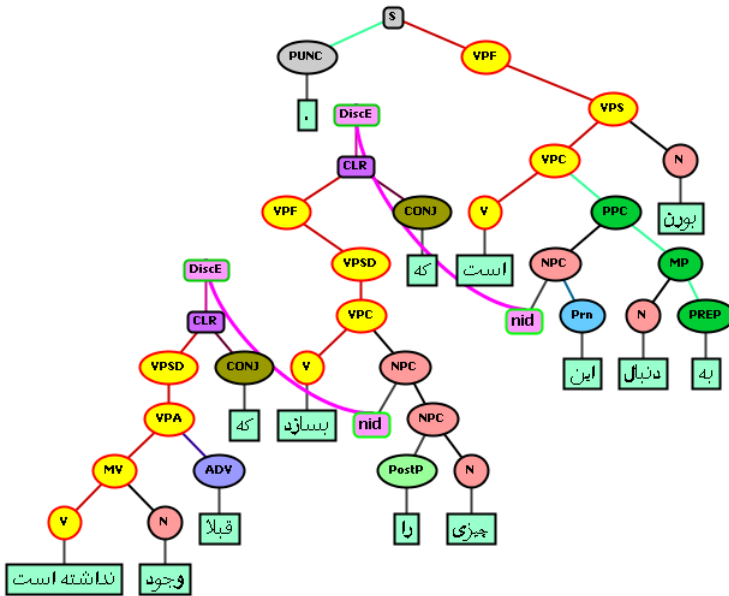


Fig. 1. Right-to-left tree representation of example (1)

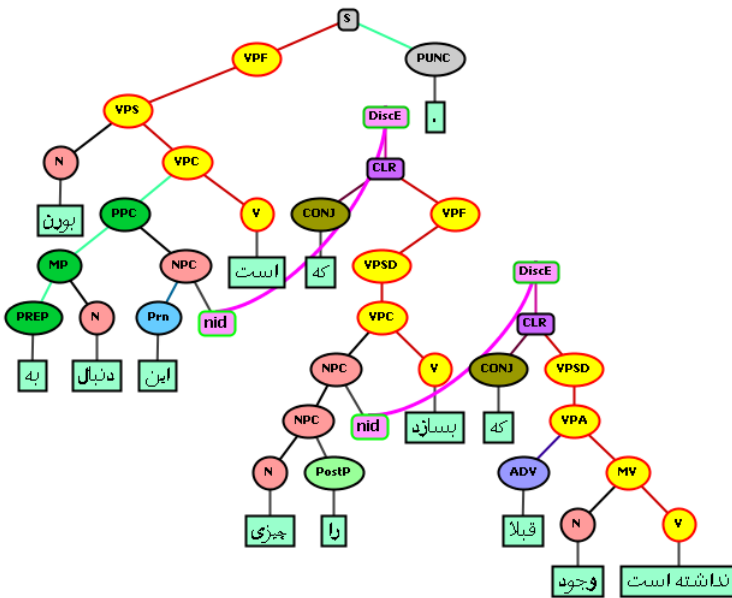


Fig. 2. Left-to-right tree representation of example (1)

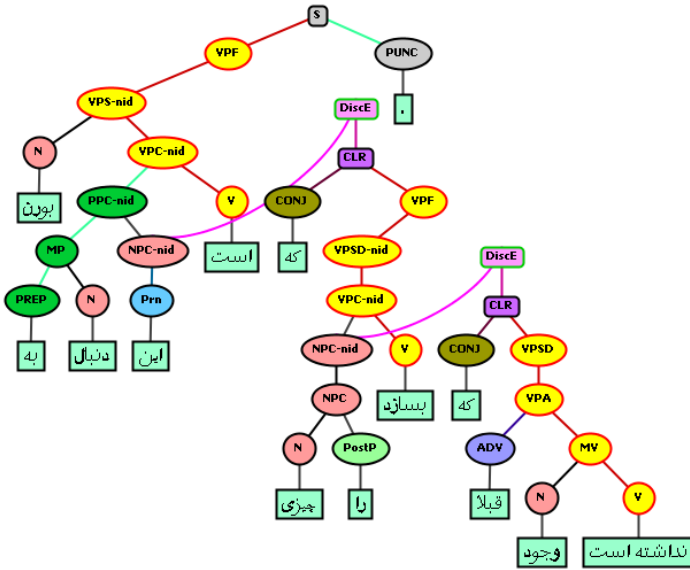


Fig. 3. Traceless left-to-right tree representation of example (1)

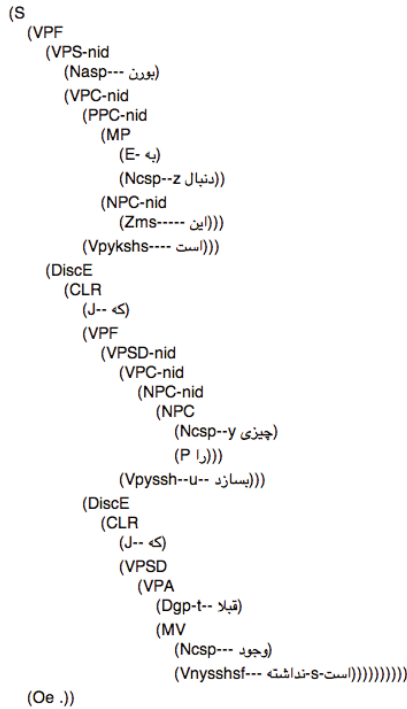


Fig. 4. Penn style tree representation of example (1)

a right-to-left language. Since the Stanford parser does not support bidirectional parsing, we have to convert the treebank into left-to-right direction, similar to Penn Arabic Treebank³, without losing any information as it is displayed in Figures 2. Additionally, since we want to train the parser with trace-less trees, the *nid* nodes should be removed. Before then, the mother nodes of the *nid* nodes are renamed as *X-nid*, and ‘-nid’ which functions as a slashed element in HPSG is propagated to the node where it is bound off by head-filler schema as represented in Figure 3. After converting the XML format of the trees into a plain text Penn Treebank format, which is demonstrated in Figure 4, the trace-less, Penn-Treebank-style data is used to train the Stanford parser for Persian. Moreover, in the normalization process, the following information is also lost from the original treebank: structure sharing, the links of the extraposed or scrambled elements to their corresponding canonical positions, the *Pragmatic* node, the named entities tags, the lemmas, and the types of /ke/, /ye/, and clitics.

After converting the original data into the Penn Treebank format, we use the *PennTreebankTokenizer* module in the Stanford parser to tokenize the input data. It is presumed that the input data is properly tokenized with a white-space. Since there is a possibility to use a white-space or pseudo-space between the elements of a word, it is replaced by ‘-s-’ in the internal structures of the lexical items to recognize multi-tokens as one unit and to solve the problem of tokenization.

We need to point out that even though the Stanford parser is the implementation of PCFG and the Persian data set used for training the parser is HPSG-based, there is no conflict between them, since the trees look like the phrase structure trees. Moreover, training a PCFG parser with an HPSG treebank is also experimented in other studies [27].

5 Class-Based Parsing

Brown [4] was the pioneer to use word clustering for language modeling methods. Later on, word clustering has been widely used in various natural language processing applications including parsing [5,6,7,16], word sense disambiguation [18], automatic thesaurus generation [13], machine translation [26], sentence retrieval [21], named entity tagging [20], language model adaptation [15], speech recognition [24], query expansion [1], and text categorization [8].

Using word clustering has advantages and disadvantages. One of the advantages of word clustering is reducing the data sparsity problem. Hence, if the word is not seen but its class, then the performance of the system will not be reduced due to the out of vocabulary problem. This approach is very effective, specially when the genre of the data changes. Another advantage of word clustering is its flexibility to capture different features. For example, semantic or syntactic properties of words can be captured using different word clustering algorithms. Since our aim for statistical parsing is to group the words with similar

³<http://www.ircs.upenn.edu/arabic/>

syntactic behavior, this flexibility gives us the opportunity to choose a sophisticated algorithm which captures the syntactic similarities of words to be used for parsing. The disadvantage of clustering is that different syntactic behaviors of homographs are not distinguished, since they are grouped in one cluster. This problem might have a counter effect for applications like parsing. Although a soft clustering approach sounds a good solution to overcome this problem, it has been shown that the overall performance of hard clustering is still better than soft clustering [10]. To resolve the problem of mis-clustering of homographs by using a hard clustering approach, we extend the word clustering algorithm by adding the POS tags of the words as an additional lexical information to the lexical items in order to recognize homographs distinctly.

Assuming that the word clustering algorithm has clustered the words of a text accurately, it is obvious that there is a clear relationship between the words belonging to the same cluster. The followings show some examples of word clusters created by the Brown algorithm [4] for Persian:

- CLUSTER1: *porxatartarin* [the most dangerous], *šomālitarin* [the most Northern], *zayiftarin* [the weakest], ...
- CLUSTER2: *pākizegi* [cleanness], *bastani* [ice-cream], *zibāyi* [beauty], ...
- CLUSTER3: *farmude?id* [have prescribed], *kardeast* [has done], *kardeand* [have done], ...

Such word clusters help us to find the set of terms syntactically related to each other. So that, if only one of the words of a cluster appears in the training data, the statistical parser can parse the input sentences which contain other words of the same cluster, even though these words do not exist in the training data. For example, if the word ‘*porxatartarin*’ has been seen in the training data and it creates a noun phrase with the term ‘*masir*’ [path], the class-based model is able to parse sentences that contain the word ‘*šomālitarin*’ which is unseen in the test data but belongs to the same cluster as ‘*porxatartarin*’, and it can be combined with the term ‘*masir*’ to create a constituent.

In the word-based scenario, the parser will be trained with the treebank containing the words with their corresponding POS tags, and the syntactic annotations. In the class-based approach, the words should be clustered into a set of predefined number of clusters. Having a mapping between the words and their corresponding clusters, the parser is trained with word clusters in the treebank instead of the words themselves.

5.1 The Brown Clustering Algorithm

Brown clustering [4] is a hierarchical bottom-up algorithm which uses Average Mutual Information (AMI) between the adjacent clusters to merge cluster pairs. Using AMI, the algorithm considers the context information to find the similar words and put them in the same cluster. To this aim, a set of word bigrams $f(w, w')$ from an input corpus is required, where $f(w, w')$ is the number of times the word w' is seen in the context w . Both w and w' are assumed to come from

a common vocabulary. Using this algorithm for clustering words, different words seen in the same contexts will be merged, because appearing in the same context shows that these words can be replaced by each other and they are assigned to the same cluster as a result [22].

One of the advantages of the Brown algorithm is using mutual information as a similarity measure. Since word bigram statistics are useful for syntax similarity, this model can be used for clustering in parsing. The mutual information of the two adjacent clusters $(C_w, C_{w'})$ is calculated as follows:

$$MI(C_w, C_{w'}) = \log \frac{P(C_w, C_{w'})}{P(C_w) * P(C_{w'})}$$

If w' follows w less often than we expect on the basis of their independent frequencies, then the mutual information is negative. If w' follows w more often than we expect, then the mutual information is positive [4]. Algorithm 1 shows the Brown word clustering algorithm in more detail.

Algorithm 1. The Brown Word Clustering Algorithm

Initial Mapping: Put a single word in each cluster
 Compute the initial AMI of the collection
repeat
 Merge the pair of clusters which has the minimum decrement in AMI
 Compute the AMI of the new collection
until reach the predefined number of clusters
repeat
 Move each word to the cluster that offer the highest AMI
until no change is observed in AMI

As shown in the algorithm, clusters are initialized with a single term in each cluster. Then, in each iteration, the best cluster pair, which offers a minimum decrement in AMI, is combined together. The process continues for $V - K$ iterations, where V is the number of terms and K is the predefined number of clusters. In the final step after the iterative process, all words are temporary moved from one cluster to the other cluster one by one, and AMI is recalculated. If this reassignment increases AMI, then the word will be moved to a cluster which offers the highest AMI. The algorithm is stopped when no additional increment in AMI is observed [4].

5.2 Word Representation for Clustering

As described in the previous section, the Brown algorithm originally used the word bigrams from a raw corpus for clustering (thereafter we call it Model A). The output of the clustering is hard; i.e. each lexical item is assigned to only one cluster. The advantage of this clustering is reducing the data sparsity which has a positive impact on statistical parsing. However, the main shortcoming of hard clustering is restricting each lexical item to one class which is not ideal for homographs. This problem is more pronounced for Persian text processing

since short vowels are not written. Bijankhan et al. [3] have defined syntactic patterns to distinguish Persian homographs; therefore, we used the POS tags of the words to disambiguate a large portion of homographs for clustering (thereafter we call it Model B). As an example, the string ‘š.v.m’ could be either pronounced /šum/ as an adjective which means ‘evil’ or /šavam/ as a verb which means ‘become.1SG’. Using the normal word clustering, these two words are treated equally, and they are assigned to only one cluster. While in the extended version, the main POS tag of the word is used as an additional lexical information for clustering; as a result, the homographs which have different POS tags are assigned to different clusters, in case they have different POS tags. To prepare the input corpus for the extended model as the input data to the Brown algorithm, the POS tag of the word is joined to the word with a hyphen, like: ‘šum-ADJ’, and ‘šavam-V’.

6 Evaluation

6.1 Setup the Experiments

Clustering Tool. Before evaluating the class-based model, we had to cluster lexical items by the Brown word clustering algorithm described in Section 5. To this aim, we used the SRILM toolkit [25] as it contains the implementation of the Brown algorithm.

Clustering Data Set. To set up the experiments of our proposed models for parsing, we used the Bijankhan Corpus for both models of clustering. The Bijankhan Corpus is a sub-corpus of Peykare, a big balanced corpus for Persian [23]. The Bijankhan Corpus contains more than 2.5 million word tokens, and it is POS tagged manually with a rich set of 586 tags containing morpho-syntactic and semantic information. Following the EAGLES guidelines [17], there is a hierarchy on the assigned tags such that the first tag expresses the main syntactic category of the word followed by a set of morpho-syntactic and semantic features. The main POS tag of the word in the Bijankhan Corpus which is a set of 14 labels is used for distinguishing homographs.

6.2 Results

To evaluate the performance of the Stanford parser for Persian based on our models, the parser is trained with PerTreeBank represented by either words or clusters. For class-based models (Models A and B), the treebank is converted in such a way that the words of the treebank are mapped to the clusters in Model A, and again the words of the treebank are mapped to the clusters with respect to their POS tags in Model B. Since no gold standard data is available for Persian, we used a 10-fold cross-validation to evaluate our models and study the impacts of our models on the parser’s performance. As a result, 10% of the data was recognized as the test data and the rest as the training data.

Table 1. The performance of the Stanford parser for clustering parsing (Model B)

Number of Clusters	F ₁ -Score
100	55.80
500	55.59
700	59.32
1000	55.81

Table 2. The performance of the Stanford parser for different models of parsing

Model	Precision	Recall	F ₁ -Score
Word	50.16	49.96	50.05
Class (Model A)	58.52	58.48	58.50
Class (Model B)	59.31	59.32	59.32

For all the experiments, a vocabulary of 90,901 terms are used for Model A, and 98,659 terms for Model B. As the two vocabulary sizes show, around 7,758 more terms are added to the vocabulary for Model B which obviously indicates that our proposed extended model has made homographs to be distinct.

Since the Brown algorithm requires a pre-defined number of clusters, we performed our experiments on 100, 500, 700, and 1000 clusters of the vocabulary terms. Table 1 presents the performance of the class-based parsing using the Model B with different numbers of clusters. As we can see in this table, the performance of the parsing is not very sensitive to the number of clusters which shows that it is not required to fine tune the number of clusters, and we can achieve a reasonable performance by different number of clusters. Nonetheless, according to the experimental results, the best performance is achieved by clustering all vocabulary terms into 700 clusters. As a result, this number of clusters is fixed for the rest of our experiments.

Table 2 compares the results of the class-based parsing (Model A and Model B) with word-based parsing. As shown in the table, the class-based models outperform the base-line word-based model. The difference between the performance of these two models are statistically significant according to the 2-tailed t -test ($p < 0.01$). This result indicates that even though the class-based approach generalizes the word representation, it has a positive impact on the performance of statistical parsing by reducing the data sparsity and solving the out of vocabulary problem. Moreover, the proposed extension of clustering (Model B) outperforms Model A which shows that adding POS information can improve the class-based parsing result by assigning homographs to different clusters. Based on the results, resolving the problem of clustering the homographs does have a positive impact in parsing such that the achieved improvement by Model B is statistically significant ($p < 0.01$) according to the 2-tailed t -test. According to the results summarized in Table 2, we can see the same behavior on the precision and recall of the models; i.e., precision and recall of the class-based models are higher than the word-based model and Model B performs the best.

7 Summary

Statistical parsers are trained with syntactically annotated data like a treebank. Not all languages have such a rich language resource for parsing; or if one exists, it suffers from the data sparsity problem. Word clustering is a recognized method for reducing the data sparsity problem and making it genre independent, since a more coarser level of the lexicon rather than the words are created. The Brown algorithm measures the syntactic similarities in a raw text to cluster words into a pre-defined number of clusters. The result of this algorithm is a hard clustering; therefore, each word is assigned into one cluster. The problem of this clustering method is that homographs are treated equally, and they are assigned into one cluster. This problem is more pronounced in Persian in which short vowels are not written. To resolve the problem relatively, we used the POS tags of the words as an additional lexical information to differentiate the homographs. We found that the class-based parsing, in general, outperforms word-based parsing significantly. Additionally, the extended model of word clustering which uses the POS tags as an additional lexical information significantly outperforms the word clustering model which uses words only.

Acknowledgement. Masood Ghayoomi is funded by the German research council DFG under the contract number MU 2822/3-1.

References

1. Aono, M., Doi, H.: A Method for Query Expansion Using a Hierarchy of Clusters. In: Lee, G.G., Yamada, A., Meng, H., Myaeng, S.-H. (eds.) AIRS 2005. LNCS, vol. 3689, pp. 479–484. Springer, Heidelberg (2005)
2. Bijankhan, M.: The role of corpora in writing grammar. *Journal of Linguistics* 19(2), 48–67 (2004)
3. Bijankhan, M., Sheykhzadegan, J., Bahrani, M., Ghayoomi, M.: Lessons from building a Persian written corpus: Peykare. *Language Resources and Evaluation* 45(2), 143–164 (2011)
4. Brown, P.F., de Souza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. *Computational Linguistics* 18, 467–479 (1992)
5. Candito, M., Anguiano, E.H., Seddah, D.: A word clustering approach to domain adaptation: Effective parsing of biomedical texts. In: *Proceedings of the 12th International Conference on Parsing Technology*, pp. 37–42 (2011)
6. Candito, M., Crabbe, B.: Improving generative statistical parsing with semi-supervised word clustering. In: *Proceedings of the 11th International Conference on Parsing Technologies*, Paris, France, pp. 138–141 (2009)
7. Candito, M., Seddah, D.: Parsing word clusters. In: *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, Los Angeles, California, pp. 76–84 (2010)
8. Chen, W., Chang, X., Wang, H., Zhu, J., Yao, T.: Automatic Word Clustering for Text Categorization Using Global Information. In: Myaeng, S.-H., Zhou, M., Wong, K.-F., Zhang, H.-J. (eds.) AIRS 2004. LNCS, vol. 3411, pp. 1–11. Springer, Heidelberg (2005)

9. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. Ph.D. thesis, University of Pennsylvania (1999)
10. Dhillon, I.S., Mallela, S., Kumar, R.: Enhanced word clustering for hierarchical text classification. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 191–200 (2002)
11. Ghayoomi, M.: Bootstrapping the development of an HPSG-based treebank for Persian. *Linguistic Issues in Language Technology* 7(1) (2012)
12. Ghayoomi, M.: From grammar rule extraction to treebanking: A bootstrapping approach. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, pp. 1912–1919 (2012)
13. Hodge, V., Austin, J.: Hierarchical word clustering - automatic thesaurus generation. *Neurocomputing* 48, 819–846 (2002)
14. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423–430 (2003)
15. Kneser, R., Peters, J.: Semantic clustering for adaptive language modeling. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). IEEE Computer Society (1997)
16. Koo, T., Carreras, X., Collins, M.: Simple semi-supervised dependency parsing. In: Proceedings of the ACL 2008, Colimbus, USA, pp. 595–603 (2008)
17. Leech, G., Wilson, A.: Standards for Tagsets. In: Text, Speech, and Language Technology, 9th edn., pp. 55–80. Kluwer Academic Publishers, Dordrecht (1999)
18. Li, H.: Word clustering and disambiguation based on co-occurrence data. *Natural Language Engineering* 8(1), 25–42 (2002)
19. Mahootiyan, S.: Persian. Routledge (1997)
20. Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discriminative training. In: Proceedings of NAACL-HLT, pp. 337–342. Association for Computational Linguistics (2004)
21. Momtazi, S., Klakow, D.: A word clustering approach for language model-based sentence retrieval in question answering systems. In: Proceedings of the Annual International ACM Conference on Information and Knowledge Management (CIKM), pp. 1911–1914. ACM (2009)
22. Morita, K., Atlam, E.S., Fuketra, M., Tsuda, K., Oono, M., Aoe, J.: Word classification and hierarchy using co-occurrence word information. *Information Processing and Management* 40(6), 957–972 (2004)
23. Pollard, C.J., Sag, I.A.: Head-Driven Phrase Structure Grammar. University of Chicago Press (1994)
24. Samuelsson, C., Reichl, W.: A class-based language model for large-vocabulary speech recognition extracted from part-of-speech statistics. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). IEEE Computer Society (1999)
25. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing (ICSLP) (2002)
26. Uszkoreit, J., Brants, T.: Distributed word clustering for large scale class-based language modeling in machine translation. In: Proceedings of the International Conference of the Association for Computational Linguistics (ACL). Association for Computational Linguistics (2008)
27. Zhang, Y., Krieger, H.U.: Large-scale corpus-driven PCFG approximation of an HPSG. In: Proceedings of the 12th International Conference on Parsing Technologies, pp. 198–208 (2011)

Building a Lexically and Semantically-Rich Resource for Paraphrase Processing

Wannachai Kampeera and Sylviane Cardey-Greenfield

Centre Tesnière - Équipe d'Accueil EA 2283
Université de Franche-Comté - UFR SLHS
30 rue Mégevand, 25030 Besançon Cedex, France
{wannachai.kampeera,sylviane.cardey}@univ-fcomte.fr
<http://tesniere.univ-fcomte.fr>

Abstract. In this paper, we present a methodology for building a lexically and semantically-rich resource for paraphrase processing on French. The paraphrase extraction model is rule-based and is guided by means of predicates. The extraction process comprises 4 main processing modules: 1. derived words extraction; 2. sentences extraction; 3. chunking & head word identification, and 4. predicate-argument structure mapping. We use the corpus provided by an agro-food industry enterprise to test the 4 modules of the paraphrase structures extractor. We explain how each processing module functions.

Keywords: paraphrase, paraphrase structures, paraphrase structures extraction, lexical resource.

1 Introduction

Paraphrase processing is an important issue in Natural Language Processing. A great number of natural language applications integrate paraphrase processing modules to improve their performance [1]. [2] provides an extensive survey on paraphrase processing.

We have carried out a linguistic analysis on a collection of 899 verbatims or emails (≈ 2376 sentences) in French in the agro-food industry domain to discover linguistic properties of paraphrases. The study reveals that a great number of paraphrases are constructed by means of derived adjectives and nouns with triggered syntactic transformations.

Inspired by the result of this linguistic analysis and by the fact that the lexicon largely varies from one domain to another whereas the sentence structure merely changes, our intention is to build a paraphrase structures database for various paraphrase processing tasks. To do so, automation guided by linguistic knowledge appears to be one of the most convenient means for extracting paraphrase structures.

In the next section, we summarize the paraphrase structures extraction model and interactions between its four main components. Following the processing flow, we explain how derived words can be extracted from Wiktionary in section 2.1.

Section 2.2 discusses the candidate sentences extraction process. In section 2.3, we provide examples of sentences output by a chunker. We describe the mapping from chunked sentences to predicate-argument structures using ontologies in section 2.4.

2 Paraphrase Structures Extraction

The architecture of the extractor and its processing flow is shown in Fig. 1. We have implemented a prototype for each of the main processing modules. For a given verb, the system outputs a set of paraphrase structures. As we want to construct a reliable lexical resource which will be used by many paraphrase processing systems, human intervention is necessary in the final validation phase.

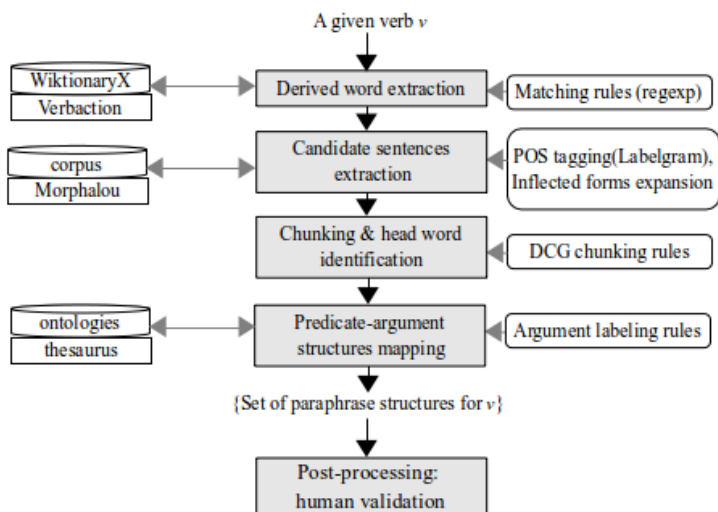


Fig. 1. Paraphrase structures extraction process

2.1 Extracting Derived Nouns and Adjectives

The first step is to identify for a given verb all its derived nouns and adjectives. There are few available French lexical resources for such a task. Verbaction [3] is an exploitable resource which provides, for a given verb v , a corresponding noun meaning *the action of v*.

As a result, our idea is to extract more derived nouns and adjectives of a given verb from WiktionaryX [4], a more compact and exploitable version of French Wiktionary. However, a lexical entry in WiktionaryX does not contain a ready-to-use derived words list, i.e. it includes antonyms, words or expressions whose meaning differs considerably from the base verb. The entry *gôûter* ‘to taste’ has the following derived words:

- *gôûteur* ‘taster’
- *dégouter* ‘to disgust’.

Being a verb, *dégouter* can easily be excluded from the candidates list. For *gôûteur* ‘taster’ however, morphological checking is needed to make sure that it is a possible derived word of *gôûter* ‘to taste’. Thus, *gôûteur* ‘taster’ will be validated if the following definition is satisfied: a noun or adjective w is said to be derived from a verb v if and only if w can be decomposed into two parts: the morphological root of v and a noun or adjective suffix.

The noun *gôûteur* ‘taster’ is decomposed into: morphological root *gout* + noun suffix *eur*. Thus, *gôûteur* ‘taster’ is accepted as a derived noun of *gôûter* ‘to taste’ and other irrelevant lexical units are rejected.

Nevertheless, the most frequent case is that Wiktionary does not provide the derived words section, hence neither does WiktionaryX. To discover missing derived words of an entry, we make use of available information such as its definition and etymology. Let us take the example of *aimer* ‘to like’ for which a semantically derived noun *amateur* ‘the person who likes’ is found by virtue of the latter’s etymological information: [...] *Du latin amator (“celui qui aime”)* [...] ‘From latin amator (“the person who likes”)’. The definition “*celui qui aime*” ‘the person who likes’ is matched by the rule which states that: for a given verb v , find its derived agent noun(s) having the meaning/definition “*person who conjugated-v*”.

The advantage of this method is that we retrieve not only lexically derived words but also those semantically derived from the verb. In fact, trying to derive *amateur* from *aimer* will not succeed because they simply have different morphological roots in modern French. The input verb together with the extracted derived words constitute the canonical keywords set.

2.2 Candidate Sentences Extraction

The verb and its derived nouns and adjectives represent the canonical keywords set as explained in 2.1. We then compile all conjugated forms of the verb and all inflected forms of the derived nouns and adjectives using the inflected forms dictionary Morphalou [5]. The result is the final keywords set K including lemmas and their inflected forms.

The next step is to extract candidate paraphrase structures from a corpus. To do so, Labelgram [6] first performs POS tagging. Next, we compare each word of the sentences against the keywords in K . The sentences which contain at least one keyword belonging to K represent the set of candidate sentences to be processed in the next module.

2.3 Chunking and Head Word Identification

In this experimental stage, chunking is done by Definite Clause Grammars (DCG) with Prolog. We exclusively target verb groups, noun groups and adjective groups so as to discover the predicate-argument structure of the sentence.

In fact, we assume that arguments are *a priori* noun groups surrounding the predicate. Ontological information on each argument determines the predicate-argument structure configuration in the next processing module (see 2.4). Predicates in this context refer to keywords in K no matter their grammatical category.

Let us give some examples of chunks, output by our chunker for the set of keywords K of the verb *décevoir* ‘to disappoint’. Note that N stands for noun chunk, ADJ for adjective, V for verb; the line immediately below each example is its literal English translation.

- (a) [N je] [V suis] [ADJ déçue] par [N une tablette de chocolat ProductName]
‘I am disappointed by a bar of chocolate ProductName’
- (b) [N je] [V trouve] [N ce produit très décevant]
‘I find this product really disappointing’
- (c) [N ce produit] [V est] [ADJ assez décevant]
‘This product is quite disappointing’
- (d) [N CompanyName] [N me] [V déçoit]
‘CompanyName disappoints me’

Before the predicate-argument labeling phase, it is necessary to identify the head of each chunk and remove trivial words because we aim for the predicate-argument structure of the sentence and not the local structure of each chunk. Also, inflected words are replaced by their lemmas at this stage. For (a), (c), and (d), identifying head words is straightforward giving the following results:

- (a) [N je] [V être] [ADJ déçue] par [N chocolat ProductName]
‘I be disappointed by chocolate ProductName’
- (c) [N produit] [V être] [ADJ décevant]
‘product be disappointing’
- (d) [N CompanyName] [N me] [V décevoir]
‘CompanyName disappoint me’

If the keyword is a part of noun groups as in (b), a decomposition rule (2) applies:

- (1) removing trivial words: [N ce produit très décevant] \longrightarrow [N produit décevant]
- (2) decomposing a noun group into a noun chunk and an adjective chunk: [N produit décevant] \longrightarrow [N produit] [ADJ décevant],

yielding:

- (b) [N je] [V trouver] [N produit] [ADJ décevant]
‘I find product disappointing’

Chunking is efficient enough for the current corpus which is domain-specific containing verbatims. This can be explained by the fact that clients (senders) use a mixture of familiar and standard language. The writing style is rather direct, emotional and brief.

2.4 Mapping into Predicate-Argument Structures

In 2.3, we obtained core structures composed of a keyword and their potential arguments. To map these structures to predicate-argument ones, in other words paraphrase structures, we have created domain-specific ontologies for this corpus. For now, we focus on two categories of entities: *Company* (company, brand, product, product names, you, your, etc.); and *Client* (I, my, our, consumer, client, etc.). The predicate-argument labeling relies on these ontologies, e.g. argument1 belongs to *Company*, argument2 belongs to *Client*.

As described in 2.3, the key assumption is that arguments are noun groups surrounding the predicate. Therefore, verbs and prepositions which do not belong to the keywords set are left as such. The result of the mapping is a set of paraphrase structures formalized as predicate-arguments. This set of paraphrase structures is ‘owned’ by the lexeme *décevoir* ‘to disappoint’:

- (a) arg2(n:client) être pred(déçu) par arg1(n:product)
- (b) arg2(n:client) trouver arg1(n:product) pred(décevant)
- (c) arg1(n:product) être pred(décevant)
- (d) arg1(n:company) arg2(n:client) pred(décevoir)

The paraphrase structures above are domain-dependent, as well as the current method for argument labeling. Nevertheless, more general paraphrase structures can be obtained by simply removing ontological information, e.g. company, product, client.

Besides, instead of using domain-specific ontologies, it is possible to generalize the labels of each argument using a thesaurus, or a semantically rich lexical database such as WordNet [7]. We are currently investigating the possibility of using Wolf [8], a WordNet for French.

2.5 Related Work

The Classificatim system [9] applied a rule-based paraphrase recognition module to classify verbatims into sets of sentences which convey the same meaning. The system reports 99% of success on corpora of the agro-food domain (with normalized text and 84% with raw text). However, writing paraphrase rules manually required a significant time and effort. The present methodology would contribute to reduce time and human effort in the rules-writing phase for such projects.

DIRT [10] is an unsupervised paraphrase extraction algorithm using the Distributional Hypothesis [11]. We think that the Distributional Hypothesis is certainly a convenient theory for simpler NLP tasks such as POS tagging. However, stating that words which occur in the same contexts tend to have similar meaning is not necessarily true. In fact, for a frequent path *I...Verb...Pizza* in our corpus, *Verb* can be anything (like, dislike, ordered, finished, throw). Secondly, while DIRT mainly extracts paraphrase patterns with two arguments *X...Y*, our approach is not concerned by such a limit because instead of using two arguments (contexts) *X...Y* to discover paraphrases, we use lexical relations such as

semantic derivation (and we will be using in future work, synonymy, negation on antonyms, hyperonymy and the likes) to find paraphrase candidates and their arguments.

3 Conclusions

We have presented, as the result of a refined linguistic analysis, a methodology for building a lexical resource for paraphrase processing. The paraphrase structures extracting method is lexical-driven and rule-based. The outlined methodology would allow rapid and innovative lexical resources' development as linguists are concerned in validating the final output by the extractor. The linguistic knowledge discovery (paraphrase structures) is mainly performed by the extractor.

References

1. Kampeera, W., Cardey, S.: Paraphrases in Natural Language Processing. In: Proceedings of the 12th International Symposium on Social Communication - Comunicación Social en el Siglo XXI, vol. II, pp. 963–967. Santiago de Cuba, Cuba (2011)
2. Androutsopoulos, I., Malakasiotis, P.: A Survey of Paraphrasing and Textual Entailment Methods. *Journal of Natural Language Processing* 11, 151–198 (2009)
3. Hathout, N., Namer, F., Dal, G.: An Experimental Constructional Database: The MorTAL Project. In: Boucher, P. (ed.) *Many Morphologies*. Cascadilla, Somerville (2002)
4. Sajous, F., Navarro, E., Gaume, B., Prévot, L., Chudy, Y.: Semi-automatic Endogenous Enrichment of Collaboratively Constructed Lexical Resources: Piggybacking onto Wiktionary. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) *IceTAL 2010*. LNCS, vol. 6233, pp. 332–344. Springer, Heidelberg (2010)
5. Romary, L., Salmon-Alt, S., Francopoulo, G.: Standards going concrete: from LMF to Morphalou. In: *Workshop on Electronic Dictionaries, Coling 2004*, Geneva, Switzerland (2004)
6. Cardey, S., Greenfield, P.: Disambiguating and Tagging Using Systemic Grammar. In: *Proceedings of the 8th International Symposium on Social Communication*, pp. 559–564 (2009)
7. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge (1998)
8. Fišer, D., Sagot, B.: Combining Multiple Resources to Build Reliable Wordnets. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) *TSD 2008*. LNCS (LNAI), vol. 5246, pp. 61–68. Springer, Heidelberg (2008)
9. Cardey, S., Greenfield, P., Bioud, M., Dziadkiewicz, A., Kuroda, K., Marcelino, I., Melian, C., Morgadinho, H., Robardet, G., Vienney, S.: The Classification Sense-Mining System. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) *FinTAL 2006*. LNCS (LNAI), vol. 4139, pp. 674–683. Springer, Heidelberg (2006)
10. Lin, D., Pantel, P.: Discovery of Inference Rules for Question Answering. *Natural Language Engineering* 7(4), 343–360 (2001)
11. Harris, Z.: *Distributional Structure*. In: Katz, J.J. (ed.) *The Philosophy of Linguistics*, pp. 26–47. Oxford University Press, New York (1985)

Tagset Conversion with Decision Trees

Bartosz Zaborowski¹ and Adam Przepiórkowski^{1,2}

¹ Institute of Computer Science, Polish Academy of Sciences

² University of Warsaw

{bartosz.zaborowski, adamp}@ipipan.waw.pl

Abstract. This paper addresses the problem of converting part of speech – or, more generally, morphosyntactic – annotations within a single language. Conversion between tagsets is a difficult task and, typically, it is either expensive (when performed manually) or inaccurate (lossy automatic conversion or re-tagging with classical taggers). A statistical method of annotation conversion is proposed here which achieves high accuracy, provided the source annotation is of high quality. The paper also presents an evaluation of an implementation of the converter when applied to a pair of Polish tagsets.

Keywords: morphosyntactic annotation, part of speech tagsets, decision trees.

1 Introduction

Most annotated corpora use various types of tags to encode additional information along words. Depending on the language and requirements they can be just parts-of-speech (POS-tags) or they can contain more complex morphosyntactic information. The sets of tags used in various corpora usually differ. They quite often differ even for corpora of the same language. Unfortunately, various Natural Language Processing tools tend to be tied to specific tagsets, or, at least, they require some effort and high quality resources to be able to switch to some other tagset. For these reasons sometimes there is a need to convert a corpus from one tagset to another. Usually, the conversion has to be automatic, since manual (re-)annotation is expensive. As in the general tagging problem, high quality results are expected, which makes the task of tagset conversion difficult.

There are two usual approaches to the automatic conversion of a corpus from one tagset to another. The first, very common, is to use a state-of-the-art tagger, train it on a large corpus tagged with the target tagset and then apply it to the source corpus. When the source corpus is tagged manually, the method is lossy in the sense that it does not make any use of these initial gold standard tags. The second common approach is to manually write a set of rules converting particular tags. Potentially it can be a very accurate method. However, this accuracy is very expensive. Additionally, the more the tagsets differ, the more difficult it is to write the rules.

The rule-based approach was deeply explored by Daniel Zeman (e.g. [13]), who proposed a rule-based conversion method involving an interset, a common tagset for different tagsets and languages. This paper explores the statistical approach.

2 Definition of the Task

Let us precisely define the task. Given a corpus tagged simultaneously with two different tagsets, the task is to train statistical conversion methods such that, given a word – or a segment¹ – and its context tagged with one tagset, the converter finds the best fitting tag from the other tagset. In this article the first tagset (the one which a given text is tagged with) will be called the *source tagset*, and the other – the *target tagset*. Similarly, a *source tag* for a given word will be a tag taken from the source tagset and a *target tag* for this word will be a tag from the target corpus for this word.

3 Baseline Approach

The simplest tagger, frequently used as a bootstrap or a baseline, is the unigram tagger. For a slightly different task from tagging, namely tagset conversion, we can modify the unigram tagger to make use of the information derived from source tags. Instead of computing frequencies of tags for each word from the training corpus, the algorithm computes frequencies of target tags for each source tag. On the basis of this information, the baseline algorithm assigns to a given word the most frequent target tag for the source tag of this word. If the source tag does not appear in the training corpus, the most frequent target tag in the whole training corpus is assigned.

4 Improvements

This section describes and discusses possible ways to improve the correctness of the algorithm. All ideas were tested on a conversion from the IPIPAN Corpus tagset ([6]) to the National Corpus of Polish (NKJP) tagset ([10])². These tagsets are compared in [5]; see also Appendix B. Experiments were performed on parts of the Enhanced Corpus of Frequency Dictionary of Contemporary Polish ([3]), which is manually annotated with tags from both tagsets. Unless stated differently, experiments were performed on a small part of the corpus,

¹ The term *word* is understood here as a maximal sequence of letters, digits and some punctuation marks (e.g., hyphens), i.e., “from space to space”; on the other hand, *segment* is the bit of text that’s assigned a morphosyntactic tag. Usually the two are the same, but Polish tagsets assume that some words consist of a number of segments (compare with English *don’t* sometimes split for tagging to *do* and *n’t*).

² These tagsets are also described at <http://www.korpus.pl/en/cheatsheet/> and <http://nkjp.pl/poliqarp/help/en.html>

consisting of about 30,000 segments from scientific texts. It will be called the *development* subcorpus. The experiments generally consisted in performing a 10-fold cross-validation on this subcorpus.

Most of the concepts described in this section can be adapted to various types of tagsets, not only positional (cf. section 4.3). The only requirement is that there exists a deterministic method to extract from tags different kinds of information represented in the source and target tagsets.

4.1 Choice of Classifier

The baseline algorithm described in the previous section can be seen as a 1R classification algorithm ([2]) in the case when there is only a single attribute (the source tag) and a class attribute (the target tag) and thus the selection of the best attribute is trivial. The 1R algorithm gives quite good results for typical data. However, it is commonly known that there exist a number of more complex classification algorithms which achieve better results using more than one attribute. As a starting point for comparison between various classifying algorithms we decided to use a small set of attributes which intuitively may contain additional information, and thus, improve the performance. The selected attributes are: a positionally encoded source tag of the word immediately preceding the given word, a positionally encoded source tag of the word and a positionally encoded source tag of the word immediately following the given word. The class attribute remains the same (a plain target tag). In the case of the current tagsets we get a class and a set of $3 * 13$ attributes, most of which are *NULL* (lack of value of grammatical categories of words which do not have those categories and do not inflect for them). The positional encoding and the context are discussed in more detail in the following two sections.

A number of experiments were performed using different classifiers from the WEKA ([11]) data-mining library. Due to performance reasons, a rough classifier selection was performed on a smaller subcorpus of approx. 5K segments (a part of the 30K development corpus).

The actual number of classifying algorithms tested amounted to 45 (all the available and applicable classifiers from WEKA, except for meta-classifiers), with over 400 different configurations. The best performing and simultaneously relatively fast four classifiers were: DecisionTable, PART, J48 and J48graft. They achieved from 89.0% to 90.2% of correctness in a reasonable computing time of at most a few minutes. After the second, more fine-grained comparison on the whole 30K development corpus, the J48 algorithm was chosen. Actually, since the J48 is a java implementation of the C4.5 ([7]) algorithm, and due to performance reasons, in later experiments we used an improved version of the original native implementation: the C5.0.

4.2 Context

The use of context is one of the most obvious improvements in tagging. It is also one of the most intuitive improvements with regard to the task of tagset

conversion. However, it is never known in advance how large the context should be for the best results on a particular language and tagset. Hence, we performed a number of experiments using various context sizes and independently changing the left context size and the right context size starting from zero (no context at all) up to 3 preceding/following segments. Like in the previous subsection, the only information available for the classifier were positionally encoded source tags for each word of the context. For the 30K development subcorpus results varied slightly from 91.9% to 92.4%. Not all context configurations improved the performance in comparison to the empty context which scored somewhere in the middle (92.1%). Larger right contexts tended to give worse results, as they probably introduced too much information noise. The best configuration found was: the right context of one segment and the left context consisting of three segments.

It is worth noting that some of the ideas described in the following sections interfere with the attributes used in these experiments. The final best configuration with all those changes is described in Section [4.7](#).

4.3 Positional Tags

Both tagsets assumed here are positional. Since both contain a large number of tags, it is likely that not enough instances of each tag will be found to successfully train any statistical methods. Even in the relatively large corpus (>300K tokens) described in the evaluation section, not all of over one thousand possible source tags appear even once. It gets even more complicated when there is a need to extract rules from the context – there sometimes are thousands or even millions of bi-grams or trigrams to cover every possible combination of tags representing a simple relation between words. To overcome this problem, source tags are split into multiple attributes: one attribute for the grammatical class and a separate attribute for each of the grammatical categories represented in the tagset.

Surprisingly, this idea is only partially profitable. The positional encoding of tags for words in the context is clearly more profitable in terms of the correctness. However switching back to the full-tag for the currently tagged word (preserving positional encoding of the context) produces better results. Especially, when the context set to empty, positional encoding of the currently tagged word causes the converter to perform worse. It turns out that separating tags deprives the classifier of useful knowledge about relations between grammatical categories. Furthermore, we observed the same bad influence on performance when the target tags were split and each grammatical category was classified separately. This negative impact of positional encoding is more visible on the small 30K development subcorpus, but can be also observed on much larger data.

4.4 Retrieving Information from the Orthographic Form and the Lemma

The most useful information available to the converter is contained in the orthographic form of a word. Unfortunately, the statistical method does not allow to

use this information directly due to data sparsity. However, some parts of the orthographic information can still be extracted. For inflectional languages such as Polish a lot of the morphological information can be obtained by analyzing only a small prefix and suffix of the orthographic form. The same goes for the lemma of words (since the source corpus is annotated manually, it can be assumed that it is also lemmatized). Another useful type of information which can be easily extracted is whether the word starts with a capital letter or not. The first idea was to extract such prefixes, suffixes and the-first-letter-cases from the word and from each of the words in the context. After some tests we narrowed down the extraction of prefixes and suffixes to one word only (without the context). The gain from including prefixes and suffixes of words from the context was not clear (in some cases it decreased correctness). It seems that for corpora of sizes similar to the development corpus (and even for larger data) prefixes/suffixes introduce too much information noise and hardly ever point at useful information. Another finding was that there is no single best prefix size or suffix size for all words. The best results can be obtained by using a few classification attributes with extracted prefixes/suffixes of different length and leave the choice of length in particular cases to the classifier.

During the tests we found out that the best configuration for the conversion task is to include the-first-letter-case for a given word and each word from the context and to extract a single-letter prefix and one-, two- and three-letter suffixes of the orthographic form of the word. Including prefixes and suffixes of the lemma didn't seem to improve results of the experiments.

4.5 Error Driven Learning of Additional Attributes

In most languages there are words which are grammatical exceptions, hard to handle by general rules. As mentioned in the previous subsection, the orthographic form or the lemma cannot be used directly to distinguish such words and treat them separately because of limited resources. However, since only a small percentage of all words behave like this, we may treat differently only such words. This leads us to the question of how to find such words. The answer is simple: the conversion algorithm can be used to find a list of words whose tags are classified incorrectly. It is done by conducting a cross-validation of the converter on training data. Of course, the converter in such a case uses only those improvements which are described in the previous sections. The set of the most frequently appearing words from this list is a good approximation of the relevant set.

Note that the set constructed this way may also contain words whose tags are frequently incorrect because of their frequent adjacency to a grammatically exceptional word, even if this exceptional word is easy to tag (e.g., because it has just one interpretation in the lexicon). In order to force the classifier to take such situations into consideration, the classifier should memorize also the context for each of the words from the set. Then, for each position in the context, it should prepare a similar set of the most frequent words at this position.

Finally, after some experiments, the classifier was enlarged with a pair of attributes for each of the words in the context each signifying “the orthographic form / the lemma is the context of a special-treatment word X” or “neither orthographic form nor the lemma is an expected context of a special word”. Of course, also a pair of attributes marking whether the given word is a “special treatment word” was added. Additionally, it appeared that it is also worth to store single-letter prefixes and suffixes of the lemmata of “special treatment” words in separate attributes. The optimal solution seem to be to use only the most frequent 1/3rd of the list of problematic words for preparing the attributes.

4.6 Using Information from Morphological Analyzer

The algorithm we describe is guessing tags for all given words. Although by design it cannot produce illegal tags (according to the tagset), there are rare cases when the classifier selects a tag not possible for a given segment. If a comprehensive morphological analyzer using the target tagset is available, it can help the classifier to overcome this problem. In order to correct such cases, a frequency table of target tags is prepared on the training data. Then, when such a case occurs, the classifier result is replaced with a tag from the set of tags proposed by analyzer which has the highest frequency.

The impact of this modification certainly depends on the quality and the size of the dictionary used by the morphological analyzer. In our case the Morfeusz SGJP analyzer ([12,9]) is used. It covers approximately 98.5% of the 30K development subcorpus and on this data only a slight improvement (about 1% reduction of the number of errors) was observed.

4.7 Fine-Tuning of Parameters

All the improvements proposed above interfere with each other. Although all of them give a correctness gain even when used together, the parameters found to be optimal for individual optimizations do not have to be optimal when all optimizations are applied at the same time. Hence, we performed another set of experiments to fine-tune various parameters. As opposed to the previous tests, here the evaluation was carried out on different sizes of corpora: small (5K tokens), medium (30K tokens) and large corpora (120K tokens). Like the development subcorpus, all of them are parts of the manually annotated Enhanced Corpus of Frequency Dictionary of Contemporary Polish. The small and medium corpora consisted of scientific texts, the large one contained also some news and fragments of plays/dramas.

For each corpus size, we performed a number of tests for slightly changed parameter values and observed for which parameters the best values change between corpora. As supposed, slight trends appeared for the context size and length of optimal suffixes, and more surprisingly, for the percentage of the most frequent problematic words used to calculate “special-treatment” attributes. All of them rose together with the corpus size. Finally, the best configuration of classifier attributes found was:

- left and right context sizes: 1 word
- different context for the “special treatment” attributes: 2 words left, 1 word right context; used for memorizing whole words (orthographic form or lemma)
- suffixes for the orthographic form: one-, two- and three-letter
- no prefixes for the orthographic form at all
- “special treatment”: single-letter prefix and suffix for the lemma of a word, two-letter suffix of the orthographic form of the processed word.
- the most frequent half of the list of the problematic words used for preparing “special treatment” attributes

5 Influence of Various Improvements on Performance

A set of experiments was performed to evaluate how the improvements described above interfere with each other. The experiments were performed on the development subcorpus (30k tokens) using 10-fold cross-validation. Table 1 shows min/max/average correctness computed on results of experiments with particular improvements enabled or disabled. It should give a rough indication of how useful each concept is. The detailed raw results of each of the experiments are shown in the appendix A. All the improvements and their parameters were in the final state, as described in Section 4.7.

Table 1. How useful each improvement is? Resources usage applies to the whole cross-validation (time is CPU time)

improvement	enabled?	correctness (%)			time and memory used
		avg	max	min	
information from orth form	yes	94.88	95.18	94.43	61min, 206 MB
	no	93.48	95.17	90.89	21min, 138 MB
context usage	yes	94.26	95.18	91.11	52min, 180 MB
	no	94.11	95.12	90.89	30min, 163 MB
special treatment words	yes	95.04	95.18	94.72	45min, 216 MB
	no	93.32	94.93	90.89	37min, 128 MB
information from morphoanalyzer	yes	94.45	95.18	92.62	41min, 172 MB
	no	93.91	95.16	90.89	41min, 172 MB
positional encoding of source tags	context + word	94.13	95.11	90.89	39min, 151 MB
	context only	94.23	95.18	91.04	38min, 146 MB
	no	94.18	95.16	91.04	46min, 219 MB

The results in Table 1 show that the biggest amount of valuable information for Polish comes from *special treatment* words (exceptions or so). The orthographic form is also very usable, but consumes significantly more resources. Surprisingly, neither the context nor the positional encoding is very important.

6 Evaluation

The final evaluation was performed on a large part of the Enhanced Corpus of Frequency Dictionary of Contemporary Polish (ECFDCP; [14]) – on all available texts annotated manually both with the IPIAN Corpus tagset and the National Corpus of Polish tagset which were aligned at the level of segmentation. It consisted of about 377,000 segments from scientific texts, news, essays and plays/dramas. During the evaluation, the target corpus was reanalyzed morphologically by means of Morfeusz analyzer (1.52% of segments were unknown to the analyzer). 56.1% of the tokens were ambiguous or had no interpretation, and the average number of ambiguous tags per token was 4.13 (including those for which there was no tag proposed by the analyzer). Due to similar tagsets, conversion of 90.64% of tags was trivial (renaming of values for grammatical class and corresponding categories) and approximately this level of correctness can be easily achieved by general, rule-based tagset converters such as DZ Interzet ([13]). Unfortunately, there is no so called *driver* for the NKJP tagset for DZ Interzet tool, therefore there is no possibility to directly compare this approach with our approach.

The other corpora appearing in Table 2 are parts of the ECFDCP described in Section 4.7. The evaluation was made by performing a 10-fold cross-validation.

Table 2. Results of evaluation. Resources usage applies to the whole crossvalidation (time is CPU time).

algorithm	corpus	correctness			resources used
		all	ambiguous	nontrivial	
here	full 377k	96.12%	93.08%	58.54%	20h, 10.5GB of RAM
baseline	full 377k	92.42%	86.49%	19.00%	5min, 600MB of RAM
here	large 120k	95.41%	91.82%	50.95%	9h, 1.9GB of RAM
baseline	large 120k	92.85%	87.25%	23.59%	2min, 350MB of RAM
here	small 5k	94.08%	89.45%	36.74%	6min, 80MB of RAM
baseline	small 5k	92.83%	87.22%	23.38%	1min, 30MB of RAM

As can be seen, the conversion approach gives quite high correctness even with a simple baseline algorithm. The baseline even for the small corpus reaches the correctness level reported for state-of-the-art taggers trained on much larger corpora of hundreds thousands tokens (e.g. PANTERA: 92.95%, WMBT: 93.00%, evaluated on 1-million corpus by [8]). The presented “here” approach achieves significantly higher correctness than state-of-the-art taggers. Furthermore, even when trained on the 30K development subcorpus and tested on the rest of the full corpus (that is, 347K segments), this method gives 94.95% of correctness. It is still better than the correctness achieved by the abovementioned taggers trained on 1M corpus.

³ Those numbers are not strictly comparable with the results of our converter, since the mentioned taggers made use of a morphological analysis from the gold standard corpus during the evaluation.

All the above “here” results were obtained using parameters tuned for the large corpus. The evaluation was performed on a computer with a 3.1GHz AMD FX processor running a 64-bit Linux and a Ruby interpreter (version 1.9.3). The baseline, as well as parts of the improved converter (data pre-/postprocessing), were implemented in the Ruby language; the C5.0 classifier used was an original native implementation. The speed and resource usage optimization was not a concern and it possibly could be up to 2 times better.

A list of most common errors is presented in Table 3. It clearly indicates that an inconsistency of the manual annotation or a different understanding of the same tags is one of the main causes of errors. Another information hard to guess by the converter are optional grammatical categories (see appendix B for details).

Table 3. A list of most common errors from the evaluation on the full corpus, covering 17% of the total number of errors

% of all errors	source tag	target tag (gold standard)	selected tag
2.14	conj	qub	conj
1.91	adv:pos	qub	adv:pos
1.73	qub	conj	qub
1.54	qub	qub	conj
0.99	qub	adv	qub
0.88	adv:pos	adv:pos	adv
0.82	conj	conj	qub
0.63	adj:pl:gen:m3:pos	adj:pl:gen:n:pos	adj:pl:gen:m3:pos
0.61	ger:sg:gen:n:perf:aff	subst:sg:gen:n	ger:sg:gen:n:perf:aff
0.60	subst:sg:nom:f	subst:sg:nom:m1	subst:sg:nom:f
0.58	qub	qub	adv
0.55	qub	qub	adv:pos
0.55	subst:sg:nom:n	subst:sg:acc:n	subst:sg:nom:n
0.53	qub	adv	subst:sg:nom:m3
0.51	subst:sg:gen:f	subst:pl:gen:f	subst:sg:gen:f
0.50	qub	qub	subst:sg:nom:m3
0.44	num:pl:nom:m3:rec	num:pl:nom:m3:congr	num:pl:nom:m3:rec
0.44	adj:pl:nom:m3:pos	adj:pl:nom:n:pos	adj:pl:nom:m3:pos
0.42	conj	adv	adv:pos
0.42	ppron3:sg:nom:f:pri	ppron12:sg:nom:f:pri	ppron12:sg:nom:m1:pri

7 Conclusions

In this paper we showed that the task of tagset conversion – in the sense of re-tagging a corpus – can be significantly improved by using information taken from the previous manual annotation. We presented various types of information extractable from manually annotated corpus that can be used in the conversion process. A comparison of results of using each type of information revealed that the most valuable information is related to orthographic forms of words, but the

source tags from the context are also useful for the algorithm. It is clear that these two types of information are partially redundant. However, using them together can further improve the correctness achieved by the converter.

We presented results which demonstrate an advantage of the current approach over using classical taggers for the task of tagset conversion. As opposed to classical taggers, this approach performs well even when there is only very little training data available for the target tagset.

References

1. Bień, J.S., Woliński, M.: Wzbogacony korpus *Słownika frekwencyjnego polszczyzny współczesnej*. In: Linde-Usiekniewicz, J. (ed.) *Prace Lingwistyczne Dedykowane Prof. Jadwidze Sambor*, pp. 6–10. Uniwersytet Warszawski, Wydział Polonistyki (2003)
2. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11, 63–91 (1993)
3. Kurcz, I., Lewicki, A., Sambor, J., Szafran, K., Woronczak, J.: *Słownik frekwencyjny polszczyzny współczesnej*. Wydawnictwo Instytutu Języka Polskiego PAN, Cracow (1990)
4. Ogrodniczuk, M.: Nowa edycja wzbogaconego korpusu słownika frekwencyjnego. In: Gajda, S. (ed.) *Językoznawstwo w Polsce. Stan i perspektywy*, pp. 181–190. Komitet Językoznawstwa, Polska Akademia Nauk and Instytut Filologii Polskiej, Uniwersytet Opolski, Opole (2003), <http://www.mimuw.edu.pl/~jsbien/M0/JwP03/>
5. Przepiórkowski, A.: A comparison of two morphosyntactic tagsets of Polish. In: Koseska-Toszewa, V., Dimitrova, L., Roszko, R. (eds.) *Representing Semantics in Digital Lexicography: Proceedings of MONDILEX Fourth Open Workshop*, Warsaw, pp. 138–144 (2009)
6. Przepiórkowski, A., Woliński, M.: The unbearable lightness of tagging: A case study in morphosyntactic tagging of Polish. In: *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC 2003)*, EACL 2003, pp. 109–116 (2003)
7. Quinlan, J.R.: *C4.5 Programs for Machine Learning*. Morgan Kaufmann, Los Alios (1993)
8. Radziszewski, A., Acedański, S.: Taggers Gonna Tag: An Argument against Evaluating Disambiguation Capacities of Morphosyntactic Taggers. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) *TSD 2012. LNCS*, vol. 7499, pp. 81–87. Springer, Heidelberg (2012)
9. Saloni, Z., Gruszczyński, W., Woliński, M., Wołosz, R.: *Słownik gramatyczny języka polskiego*. Wiedza Powszechna, Warsaw (2007)
10. Szalkiewicz, Ł., Przepiórkowski, A.: Anotacja morfoskładniowa NKJP. In: Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B. (eds.) *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw (2012)
11. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005), <http://www.cs.waikato.ac.nz/ml/weka/>

12. Woliński, M.: Morfeusz — a practical tool for the morphological analysis of Polish. In: Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. Advances in Soft Computing, pp. 503–512. Springer, Berlin (2006)
13. Zeman, D.: Reusable tagset conversion using tagset drivers. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008. ELRA, Marrakech (2008)

A Detailed Evaluation Results

Table 4. Results for the experiments with enabling different combinations of improvements. Obtained on the 30K development corpus, values in percents. Background color reflects the results (darker – worse). Improvements names: ctx – context, pos – positional encoding, special – special treatment words, morph – morphological analyzer, orth form – information extracted from orthographic form. See Section 5 for more details.

			no orth form			orth form		
			no pos	pos word + ctx	pos ctx only	no pos	pos word + ctx	pos ctx only
special –	morph –	ctx –	91.04	90.89	91.04	94.43	94.45	94.43
		ctx +	91.11	91.25	91.40	94.56	94.50	94.56
	morph +	ctx –	92.79	92.62	92.79	94.74	94.74	94.74
		ctx +	92.85	93.02	93.16	94.85	94.84	94.93
special +	morph –	ctx –	94.91	94.72	94.91	95.12	95.03	95.12
		ctx +	94.99	94.99	95.03	95.16	95.10	95.13
	morph +	ctx –	95.00	94.77	95.00	95.12	95.01	95.12
		ctx +	95.06	95.08	95.17	95.14	95.11	95.18

B Comparison of the IIPAN and NKJP Tagsets

The tagsets used in this article were designed for two large corpora of Polish: the IIPAN Corpus of Polish and the National Corpus of Polish (NKJP). Both are positional, with quite a large number of possible morphosyntactic tags. The sets of grammatical categories and grammatical classes are similar between the tagsets. Hence, most of tags from the one tagset have a corresponding tag in the other tagset. In Tables 5 and 6 we present a comparison of grammatical categories and grammatical classes between tagsets with differences highlighted.

Some statistics: the number of all theoretically possible tags in the IIPAN tagset is 4389, 1357 of them were proposed in the lexical analysis, the number of tags actually used in the annotation is 913. The number of all theoretically possible tags in the NKJP tagset is 4187, proposed in the lexical analysis: 1697, 792 of them were actually used in the corpus annotation.

Table 5. Grammatical categories and their values in the two tagsets

grammatical category	values for IPIPAN tagset	values for NKJP tagset
number	sg, pl	sg, pl
case	nom, gen, dat, acc, inst, loc, voc	nom, gen, dat, acc, inst, loc, voc
gender	m1, m2, m3, f, n	m1, m2, m3, f, n
person	pri, sec, ter	pri, sec, ter
degree	pos, comp , sup	pos, com , sup
aspect	imperf, perf	imperf, perf
negation	aff, neg	aff, neg
accommodability	congr, rec	congr, rec
accentability	akc, nakc	akc, nakc
post-prepositionality	npraep, praep	npraep, praep
agglutination	agl, nagl	agl, nagl
vocalicity	nwok, wok	nwok, wok
fullstoppedness	(not present)	pun, npun

Table 6. Grammatical classes with their morphosyntactic characteristics in the two tagsets. Only differing classes are shown here; 28 classes have been omitted (see references cited in text for full lists). Categories listed in [] are optional, their values may be present or not. The rest is required.

grammatical class	morphosyntactic characteristics	
	categories for IPIPAN tagset	categories for NKJP tagset
num	number, case, gender, [accom- modability]	number, case, gender, accom- modability
numcol	number, case, gender, [accom- modability]	number, case, gender, accom- modability
adjc	(class not present)	(no categories)
adv	degree	[degree]
xxs	number, case, gender	(class not present)
comp	(class not present)	(no categories)
brev	(class not present)	fullstoppedness
burk	(class not present)	(no categories)
interj	(class not present)	(no categories)

Fitting a Round Peg in a Square Hole: Japanese Resource Grammar in GF

Elizaveta Zimina

University of Gothenburg, Sweden
lizazim@gmail.com

Abstract. This paper is a report on the development of a Japanese resource grammar in Grammatical Framework (GF), a programming language for multilingual grammars and their applications. Japanese became the 25th language in the GF Resource Grammar Library (RGL). Grammars in the RGL are based on a common abstract syntax, which covers grammatical categories and syntactic relations that are supposed to be shared by different languages. Being typologically and genetically distant from the languages covered by the RGL, Japanese disputes the generality of some rules in the RGL abstract syntax and brings up new issues in the discussion on the universal properties of languages.

Keywords: Japanese, Grammatical Framework, resource grammar, natural language processing, computational linguistics, machine translation.

1 Introduction

GF (Grammatical Framework [1]) is a grammar formalism for multilingual grammars and their applications. Together with implementation it forms a framework for performing various natural language processing tasks.

A GF grammar is based on an abstract syntax, which determines a set of abstract syntax trees built with a glance to the semantically relevant language structure, and one or more concrete syntaxes, which designate the way of mapping of abstract syntax trees on to strings and vice versa.

The process of producing a string from an abstract syntax tree is called *linearization*. The opposite, *parsing*, means deriving an abstract syntax tree (or several) from a string.

Expressivity of GF was studied in [2]. Its subclass named *context-free GF* is proved to be equivalent to Parallel Multiple Context-Free Grammar (PMCFG [3]). PMCFG involves a polynomial parsing algorithm and is characterized by substantial expressive power.

The GF Resource Grammar Library (RGL) [4] is a set of natural language grammars implemented in GF. These grammars are built on the basis of common abstract syntax, i.e. a common tree structure. The RGL aims at sound descriptions of natural languages in terms of the linguistics structure.

Spoken by 122 mln people and being the 9th most common language in the world [5], Japanese became the 25th language in the RGL. We do not merely

want to describe the implementation of the Japanese resource grammar; a number of publications of this kind for other languages have been already produced [8], [9], [10]. The paper is aimed to focus on some linguistic peculiarities of the Japanese language that required special treatment in linearization of abstract syntax rules.

A number of language processing tasks, such as multilingual generation, software localization, natural language interfaces, and spoken dialogue systems, can be realized by means of the RGL. GF is being successfully applied in a number of experiments within the MOLTO project [11], aimed at high-quality translation between many simultaneous languages on the web.

Japanese is rather distant both genetically and typologically from most of the languages whose grammars have already been implemented in GF [6], [7]. Therefore, the Japanese resource grammar in particular can help to define the level of universality of the RGL abstract syntax rules, i.e. to decide whether these rules are possible to be realized by linguistic means of any language, regardless of its structure and language family.

In Section 2 we summarize what grammatical categories and syntactic relations are covered by the RGL abstract syntax. Section 3 shows some challenges and inconsistencies that came up as a result of coding those syntactic relations with respect to the Japanese language and that have not been observed before in GF grammars of other languages. We describe the evaluation of the developed grammar in Section 4, view the related work in Section 5 and, finally, sum up the results of the research in Section 6.

2 Syntactic Structures of the RGL

Each word predetermined by the RGL abstract syntax belongs to one or another category, which in most cases (but not always) coincides with the corresponding conventional linguistic unit: N (noun), V (verb), S (sentence), Text, etc. In concrete syntax, these categories can have any set of features (any *record type*), depending on the grammar peculiarities of a certain language. For example, in Japanese concrete syntax the PN (proper noun) category has the following record type:

```
cat1  PN ;    -- proper name, e.g. "Paris"
lincat PN = {s : Style => Str ; anim : Animateness} ;
```

The PN record type presented above means that the choice of string representation of a proper noun depends on the speech style: we add the postfix *さん* “san” to people’s names in the respectful style (ジョンさん “John-san”). PNs also have the label *anim* (animateness) for further agreement with predicates, specifically, with the copula verbs: *いる* “iru” is used with animate subjects and *ある* “aru” – with inanimate ones.

¹ In GF notation, the reserved words *cat* and *fun* correspond to abstract syntax, *lincat* and *lin* – to concrete syntax.

Most RGL categories are assembled in the module *Cat*. Another module, *Common*, covers categories that uniformly have the linearization $\{s : \text{Str}\}$ for most languages. These include all kinds of adverbs, categories related to tense forms, embedded sentences, Phrase and its components, and Text. However, in case of Japanese, the “common” categories still have some more parameters and labels. The main reason for this is the necessity of covering both plain and polite styles, which affects practically all categories in the concrete syntax. Moreover, individual words belonging to the same “common” category often require special grammar in Japanese, that is, separate linearization structures.

Most important grammar rules are combined in the modules according to the linguistic principle. E.g. rules of forming and modification of noun phrases (NPs) are stored in the *Noun* module. There one can find rules through which nouns can be complemented with attributives, numerals, articles, adverbial constructions, other nouns as objective complements, embedded sentences and questions and so on.

Similarly, other syntactic modules for phrase building contain rules for adjectives, verbs, numerals and adverbs. A special module is devoted to structural words, such as prepositions, conjunctions, determiners, etc. Structures of a higher order are gathered in files named *Sentence*, *Question*, *Relative* (for relative clauses), *Idiom* (for common idiomatic expressions), *Phrase* and *Text*. The *Lexicon* contains a test set of 350 content words. The abstract parameters of tense, polarity and anteriority are defined in *Tense*.

The above-mentioned modules form the most important part of the RGL abstract syntax. Concrete syntax for any language should cover all the predetermined rules. Moreover, a resource grammarian should provide inflection tables for different parts of speech and other language-specific functions in *Paradigms* and *Res* (resource module). Specific linguistic phenomena, not covered by the common abstract syntax, can be stored in abstract and concrete *Extra* modules. However, these features are disregarded by the common API, which covers only the shared syntax (categories and functions for syntactic combinations) and provides multilingualism of GF grammar applications.

3 Implementation of the GF Japanese Resource Grammar

Japanese, a national language of Japan, belongs to the Japanese-Ryukyuan language family, whose genetic relations with other languages are still disputable [12]. It combines features of the agglutinative and inflectional language types. The Japanese name system is almost entirely agglutinative; grammatical relations are expressed by means of postpositive elements (postpositions, particles). Inflectional features are realized in joining word stems and affixes varying in their structure. The systems of Japanese verb and predicative adjective are considerably inflectional, though there are also a lot of formants (conjunctions, particles, etc) there [13].

The following subsections describe some peculiarities of the Japanese language, problems connected with taking these peculiarities into account in writing

GF concrete syntax and their solutions (or reasons for the impossibility of solving them). We omit a number of linguistic peculiarities that have been observed before in the RGL grammars for some languages. For example, the system of counters added to numbers for counting nouns is also characteristic of the Thai language. And the requirement to introduce a numeral unit for 10,000, alongside with usual tens, hundreds and thousands, has been already satisfied in the RGL grammars for Urdu, Hindi and Thai.

3.1 Stylistic Differentiation

The Japanese language uses a complex system of markers of polite and plain styles, depending on the politeness requirement of the situation, social standing of the speakers and the person being talked about. Politeness markers appear on verbs, adjectives, and even nouns. For example, the plain (dictionary) form of the verb to eat, 食べる “*taberu*”, is used when speaking with someone close to the speaker; but the speaker should use the politeness marker *ます* “*masu*” (食べます “*tabemasu*”) if an interlocutor is a stranger or is older than the speaker [14]. In fact, the necessity to consider both styles in the concrete GF grammar makes the paradigms of all content parts of speech twice larger. For example, the paradigm for *i*-adjectives, one of the two groups of Japanese adjectives, looks as follows:

```
i_mkAdj : Str -> Adj = \chiisai -> let chiisa = init chiisai ; in {
  pred = table {
    Resp => table {
      TPres => table {
        Pos => chiisai ++ " です" ;
        Neg => chiisa + " くありません" } ; ... } ;
    Plain => table {
      TPres => table {
        Pos => chiisai ;
        Neg => chiisa + " くない" } ; ... } ;
```

Since the uppermost GF category, Text, should be of the type {s : Str}, without any parameters in order to provide correspondence with other languages, we chose the plain style to be a “default” one. The polite form can be obtained only through the *Extra* module by means of the introduced category *Level*.

3.2 Markers of Topic and Subject

An NP representing a subject of a sentence is usually followed by the particles は “*wa*” or が “*ga*”. は is used to mark the topic of a sentence and to express contrast. が often introduces a new subject. The choice of は or が in a particular case can be complicated by questions of context and the speaker’s intent:

寿司はおいしい。	寿司がおいしい。
<i>Sushi wa oishii.</i>	<i>Sushi ga oishii.</i>
Sushi: it’s delicious.	Sushi is delicious.

In these sentences, the topic marker は directs attention forward to the predicate (*it's delicious*), and the subject particle が emphasizes what precedes it (*sushi*). It is not always easy (nor indeed necessary) to convey this distinction in an English translation [15].

In our grammar we do not carry out a contextual analysis, so we provide usage of both particles through the parameter *Particle*. Like for the *Style* parameter, we had to choose the “default” value for *Particle* – は. The special category *Part* was introduced in the *Extra* module to provide the usage of the が particle. Within the shared syntax, が is also involved in subordinate clauses.

3.3 Tense

Japanese is often described as having just two tenses: past and non-past; there is no separate future tense [16]. This means that all forms of verbs and predicative adjectives for the present and future tenses (*TPres* and *TFut*) coincide. The conditional tense also corresponds to the Japanese non-past (yet, it is usually expressed through the *ba*-form of a verb or adjective).

Japanese also does not have special forms to express anteriority. Thus, in our grammar the English future perfect is replaced with the Japanese non-past tense and the present and past perfect correspond to the Japanese past tense, although, for the present perfect, this is not always a correct decision.

In fact, 16 possible tense forms provided by the RGL common abstract syntax (4 tenses * 2 anteriority types * 2 polarity types) are reduced to 4 distinct Japanese forms ((past + non-past) * (positive + negative)).

3.4 Conjugation of Verbs and Adjectives

In Japanese concrete syntax, the above-mentioned four tense forms are realized in conjugation tables of verbs and adjectives (used in the predicative function, e.g. *the table is small*). Aside from these forms, we used a number of other verb and adjective forms peculiar to the Japanese language.

The conditional, or so-called *ba*-form expresses the meaning of the English word *if*. The *te*-form usually replaces a word in a row of homogeneous sentence parts (attributes or predicates) connected by the conjunction *and*. The *te*-form of verbs is also used to create tenses and moods, such as the continuous tenses, the imperative mood or for conjunction of clauses within one sentence. Both *ba*- and *te*-forms can be positive or negative.

These forms express grammatical relations that are normally conveyed by separate functional words in European languages. The record field of the Clause (Cl) category obtained the *ba* and *te* fields containing clauses with predicates in corresponding forms:

```
lincat Cl = {s : Particle => Style => TTense => Polarity => Str ;
            te, ba : Particle => Style => Polarity => Str ; ... } ;
```

We also had to create the field for the adverbial form of adjective, which is then inherited by the adjective phrase (AP) category, because this form of AP is used in some verb phrase (VP) constructions (e.g. *become red*, *paint (it) red*).

We have to note that not all conjugation forms of adjectives and verbs are used in our concrete syntax. By means of structural words, we avoided using some other grammatical forms that are peculiar to the Japanese language. For example, presumptive, provisional and causative forms were left aside, though they are often used in Japanese natural speech [16].

3.5 The Verb “Want”

In GF grammar the verb *want* is defined as a verb-phrase-complement verb (VV). It has a number of equivalents in Japanese. We had to include at least two of them in our grammar, as there is no universal variant suitable for all subjects. Thus, to express one’s own wish to do something, one should add たい “-tai” to the verb’s *i*-stem².

来年日本に行きたいです。
Rainen Nihon ni ikitai desu.
 [next year] [Japan] [to] [want to go]
 I **want to go** to Japan next year. [15]

The たい form conjugates in the same way as *i*-adjectives: However, if we are talking about someone else wanting to do something, then we use the ending たがる “tagaru” (or, more precisely, its continuous form -たがっている “tagatte iru”) instead of たい:

先生はコーヒーを飲みたがっています。
Sensei wa koohii o nomitagatte imasu.
 [teacher] topic marker [coffee] object marker [want to drink]
 The teacher **wants to drink** coffee.

To correlate with the subject (i.e. to check if it is the pronoun *I* or not), the categories VV and VP got the parameter *Speaker* with values *Me* and *SomeoneElse*. NP also has the corresponding feature *meaning*. At the Clause level VP agrees with NP’s *meaning* value. The verb *want* obtained a special record:

```
mkWant : VV = {
  s = table {
    Me => \\st,t,p => (i_mkAdj "たい").pred ! st ! t ! p ;
    SomeoneElse => \\st,t,p => "たがって" ++
      (mkVerb "いる" Gr2).s ! st ! t ! p} ; ... } ;
```

3.6 The Verb “Give”

Knowing whether an NP is *Me* or *SomeoneElse* is also essential for solving the problem of the verb *to give*. There are five equivalents of this verb in Japanese; their usage depends on the statuses of the giver and receiver. Schematically this is represented in Fig. 1.

² In our concrete syntax, the verb paradigm contains records for *i*- and *a*-stems used in a number of functions.

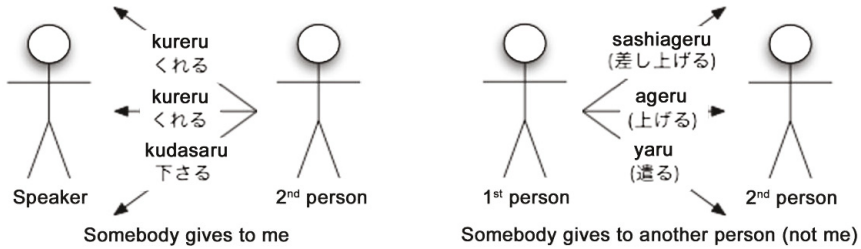


Fig. 1. Japanese Verbs of Giving [17], [18]

We simplified this grammar and took more or less neutral verbs くれる “kureru” and 上げる “ageru” for the case when the receiver is the speaker (that is, *somebody gives me*) and all other cases.

太郎は私にマンガがくれた。

Tarou wa watashi ni manga o kureta.

[Taro] topic marker [I] [to] [manga] object marker [gave]

Taro gave me a manga.

私は妹にお菓子上げる。

Watashi wa imouto ni okashi o ageru.

[I] topic marker [sister] [to] [candy] object marker [give]

I give my sister a candy.

We did not take the verbs 差し上げる “sashiageru” (*give to superiors*), 下さる “kudasaru” (*give to a speaker from a superior*) and 遣る “yaru” (*give to inferiors*; rather disrespectful) since we cannot decide whether the giver is superior or not. Thus, 先生 “sensei” (*teacher*) is likely to be a superior, but we are unable to analyse the context and decide, for example, who *John* is.

The verb *give* gets a special record in the *Res* file:

```
mkGive : Verb3 = {
  s = table {
    Me => \\st,t,p => (mkVerb " くれる " Gr2).s ! st ! t ! p ;
    SomeoneElse => \\st,t,p => (mkVerb " 上げる " Gr2).s ! st ! t ! p
  } ; ... } ;
```

When we complement a 3-place verb (V3, the category that *give* belongs to) with an NP, the correct verb is chosen depending on the NP’s *meaning* field – *Me* or *SomeoneElse*:

```
fun Slash3V3 : V3 -> NP -> VPSlash ; -- give (it) to her
lin Slash3V3 v3 np = {
  s = \\sp,st,t,p => v3.s ! np.meaning ! st ! t ! p ; ... } ;
```

3.7 Relative Clauses

In Japanese there are no relative pronouns; it is sometimes only from the context (or knowledge of the world) that one can determine the case relation between a relative clause (RCI) and the noun it modifies [16]:

僕が記事を書いたレストラン

Boku ga kiji o kaita resutoran

[I] subject marker [article] object marker [wrote] [restaurant]

(1) A restaurant *about which* I wrote an article

(2) A restaurant *in which* I wrote an article

In a number of operations with RCIs it is essential for us to know whether the subject of a RCI is represented by a sole relative pronoun or some NP. In the first case (e.g. *who is sleeping*) the subject is actually missing, but in the clauses like *whose mother is sleeping* or *whom John loves* the subject is present. To differentiate these types of RCIs we introduced the Boolean label *missingSubj*.

The correct conjunction of Japanese RCIs is possible only if subjects of all those clauses are missing. In fact, these clauses are just verb phrases that can be easily joined. The meaning of the conjunction *and* is expressed by the verb's *te*-form, so we take the *te*-form of a clause and put a comma after it. The verb of the last clause is taken in the plain form. The meaning of the conjunction *or* between RCIs of this type is expressed by the particle か “ka”, like between nouns or adjectives.

However, in Japanese it is practically impossible to join relative sentences that have subjects (e.g. *whose mother is sleeping and with whom I walk*). It is recommended to restructure these constructions and split them in two (or more) sentences. But since the RGL abstract syntax determines that *all* RCIs can join together, this inconsistency results in the ungrammatical output.

4 Evaluation of the Japanese Resource Grammar

To assess the correctness of the developed Japanese grammar we used a test set that contains syntax trees corresponding to 583 test units (words, phrases and sentences). We automatically generated English and Japanese linearizations of these trees and assessed the correctness of the Japanese translations. The test set is composed of the test units developed in [19] and examples used in the synopsis of the RGL [20].

The test set is aimed not only at proof-reading but also reflection of the resource library's coverage. It is organized in such a way that displays all the rules in the grammar and possible combinations of the categories. Test units for higher-order categories (such as Phrase or Utterance) additionally verify the usage of their constituents (NP, AP, VP, etc).

The test has confirmed our assumption that some grammatical constructions predetermined in the abstract syntax are hardly possible, if not impossible, to implement in Japanese. For example, since Japanese has no relative pronouns, it is undesirable to use complex RCIs. The following structure converted into

Japanese is formally grammatical, but its semantics is totally lost as there is no way to denote the relation between the main and subordinate clauses:

LangEng: there is a woman the mother of whom John loves

LangJap: ジョンがお母さんを愛する女はいる

[John] subject marker [mother] object marker [love] [woman] topic marker [is]

As we can see, not only a relative pronoun is missing, but also the preposition determining the relation of *mother* to *woman*. In the natural Japanese language, one would restructure this phrase and possibly split it into successive clauses.

Except of the above mentioned nuances, no serious obstacles have been revealed in the application of the Japanese resource grammar. Testing the Japanese resource grammar was an important and final step in the debugging process.

To test the parsing speed, we took 100 sentence trees from the above-mentioned test set and parsed their linearizations in Japanese and English. The general parsing time for Japanese was 4937 msec, for English – 169 msec. So on the average parsing of a Japanese sentence was 29,2 times longer, than of its English equivalent. We suppose that the main reason for this is that Japanese grammar structures are much more ambiguous, than English ones.

Consider the sentence *The woman laughs*. Its parsing in English takes 156 msec and produces 3 parse trees: two trees appear in addition to the main one because of two Sentence-level rules from the Extra module, which partly coincide with the general function.

The Japanese equivalent, 女は笑う “onna wa warau”, is parsed in 3182 msec and obtains 40 readings. Firstly, the reason for this is that the *Extra* module involves the Level category; in case of the plain style, which is set as a default for the common syntax, trees obtained through the common syntax rules are doubled by the extra rules with the plain style value.

Secondly, four tense forms instead of the available 16 forms described in subsection 3.3 are among the most evident reasons for ambiguity.

Thirdly, Japanese nouns do not change their forms to express grammatical relationships, take no articles and do not generally have plural forms. Therefore, English phrases *a woman*, *the woman*, *women*, *the women* have one and the same equivalent in Japanese – 女 “onna”. Moreover, it does not make sense to distinguish mass nouns in Japanese, since it lacks sufficient capability to singularize and pluralize nouns [21]. So, in our grammar, 女 is also regarded as a mass noun (one that needs no article).

Adverbial structures represented by subordinate clauses can also be a source for ambiguity. As well as simple adverbs, they can be attached to NP, AP or VP. Because of rather strict word order, Japanese does not allow one to attach a subordinate clause directly to the sentence member modified by it. Normally, adverbial subordinate clauses should be placed before the main clause in Japanese. Therefore, this complicates the problem of clause attachment and aggravates ambiguity.

The source code for Japanese equals 3952 lines of GF code, which is close to the average value for other languages. The complexity of the run-time grammar, due to parameters, branching, etc, can be measured by the size of context-free

approximation. Without the lexicon, the Japanese grammar expands to 19,961 context-free rules. To compare, English has 65,902 rules due to richer inflection, whereas Thai has 706 rules.

5 Related Work

The distinctive feature of GF is that it involves a mapping from the abstract syntax, which is common to all languages, to concrete linearizations, which reflect peculiarities of each language. Thus, the GF approach maintains a compositional relation among different languages.

The starting point of most previous works on building Japanese computational grammars was the Japanese linguistics as such, with its peculiarities and distinctions from grammars of the European languages.

Masuichi et al. [22] developed a Japanese parsing system involving an elaborate hand-coded grammar on the basis of the Lexical-Functional Grammar (LFG) formalism. This parser managed to cover more than 97% of real-world text. The parser based on the Japanese Phrase Structure Grammar [23] takes into account such linguistic phenomena in Japanese as word order variation, gaps in a sentence and relativization.

Some Japanese grammars were developed to parse particular kinds of text. For example, the Verbmobil HPSG grammar for Japanese [24] is designed to parse Japanese spoken language and identify spoken Japanese phenomena, such as topicalization, honorification and zero pronouns. The English to Japanese Medical Speech Translator built on the basis of REGULUS 2 [25] operates with a vocabulary of about 200 words and translates spoken yes-no questions about headache symptoms from English to Japanese.

6 Results

The main result of the conducted research is building the Japanese grammar as a part of the GF Resource Grammar Library. The grammar covers all the categories and rules of the RGL abstract syntax, thus providing the full correlation with the resource grammars of other languages in the GF library.

Testing and evaluation of the developed grammar revealed a number of inconsistencies in English-Japanese translation, which can be explained by at least two facts:

- Japanese is typologically and genetically distant from the European languages, it involves a number of special grammar structures, which are often conditioned by context.
- The RGL abstract syntax is based on the linguistic regularities, some of which are peculiar to the European languages.

Therefore, building of the Japanese resource grammar is also an experiment in the field of language learning and interlanguage comparison. The observed linguistic phenomena made us draw the following conclusions:

- Japanese has no universal grammar that suits every speech situation. At least one should operate with two styles – plain and honorific.
- In general, Japanese has enough grammatical and semantic resources to cover all the rules in the RGL abstract syntax and produce correct grammar structures, though it involves some special morphological categories that are out of scope of the RGL abstract syntax (e.g. particles), while some other categories are missing in Japanese (e.g. relative pronouns).
- Substantial obstacles in linearization of rules were observed only in one, yet important grammar segment – forming and conjoining of subordinate clauses. Complex subordinate constructions are avoided in the Japanese natural language.

The obtained results can contribute to the discussion on the universal properties of languages and propose to reconsider some rules in the RGL abstract syntax.

Acknowledgments. The present work was carried out within the Master Thesis course at the University of Gothenburg, Sweden. I want to thank the Centre of Language Technology of the University of Gothenburg for financial support and my supervisor, Professor Aarne Ranta, who provided valuable assistance and constructive criticism to the research summarised above. The work also benefited from the input of the Japanese language consultants, Viktoria Novikova and Ayako Ugamochi.

References

1. Ranta, A.: *Grammatical Framework: programming with multilingual grammars*. CSLI Publications, Stanford (2011)
2. Ljunglöf, P.: *Expressivity and Complexity of the Grammatical Framework*. PhD thesis, Computer Science, Göteborg University (2004), <http://www.cse.chalmers.se/~peb/pubs/Ljunglof-2004a.pdf>
3. Seki, H., Matsumara, T., Fujii, M., Kasami, T.: On multiple context-free grammars. *Theoretical Computer Science* 88, 191–229 (1991)
4. Ranta, A.: The GF Resource Grammar Library. *Linguistic Issues in Language Technology* 2(2) (2009), <http://elanguage.net/journals/index.php/lilt/article/viewFile/214/158>
5. Lewis, M.P. (ed.): *Ethnologue: Languages of the World*, 16th edn. SIL International, Dallas, Tex (2009), <http://www.ethnologue.com/>
6. Dryer, M.S.: *Genealogical Language List*, <http://wals.info/languoid/genealogy>
7. Eifring, H., Theil, R.: Linguistic typology. In: *Linguistics for Students of Asian and African Languages* (2005), <http://www.uio.no/studier/emner/hf/ikos/EXFAC03-AAS/h05/larestoff/linguistics/Chapter%204.H05.pdf>
8. Angelov, K.: *Type-Theoretical Bulgarian Grammar*. In: Nordström, B., Ranta, A. (eds.) *GoTAL 2008*. LNCS (LNAI), vol. 5221, pp. 52–64. Springer, Heidelberg (2008)

9. Khegai, J.: GF parallel resource grammars and Russian. In: Proceedings of ACL 2006 (The Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics), Sydney, Australia, pp. 475–482 (July 2006)
10. Virk, S., Humayoun, M., Ranta, A.: An Open Source Urdu Resource Grammar. In: Proceedings of the 8th Workshop on Asian Language Resources (Coling 2010 workshop) (2010)
11. The MOLTO project, <http://www.molto-project.eu>
12. Robbeets, M.: Belief or argument? The classification of the Japanese language. Eurasia Newsletter 8. Graduate School of Letters, Kyoto University (2004)
13. Alpatov, V.M., Arkadiev, P.M., Podlesskaya, V.I.: Theoretical Grammar of Japanese. Teoreticheskaya grammatika yaponskogo yazyka, vol. 2. Natalis, Moscow (2008)
14. Miyagawa, S.: Japanese Language. Massachusetts Institute of Technology (1999), <http://web.mit.edu/jpnet/articles/JapaneseLanguage.html>
15. Bunt, J.: Oxford Japanese grammar & verbs. Oxford University Press (2003)
16. Kaiser, S., Ichikawa, Y., Kobajashi, N., Yamamoto, H.: Japanese: a comprehensive grammar. Routledge, London (2004)
17. Wikibooks: Japanese/Lessons/Giving and Receiving, http://en.wikibooks.org/wiki/Japanese/Lessons/Giving_and_Receiving
18. Akiyama, N., Akiyama, C.: Japanese Grammar, 2nd edn. Barron's Educational Series, New York (2002)
19. Khegai, J.: Language engineering in Grammatical Framework (GF). PhD thesis, Computer Science, Chalmers University of Technology, Gothenburg (2006)
20. GF Resource Grammar Library: Synopsis, <http://www.grammaticalframework.org/lib/doc/synopsis.html>
21. Mazack, M.J.M.: A comparative analysis of noun classification in English and Japanese. Working Paper. Western Washington University (2007)
22. Masuichi, H., Ohkuma, T., Yoshimura, H., Harada, Y.: Japanese parser on the basis of the Lexical-Functional Grammar Formalism and its Evaluation. In: Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17), pp. 298–309 (2003)
23. Fujinami, Y.: An implementation of Japanese Grammar based on HPSG. Master's thesis. Department of Artificial Intelligence. Edinburgh University (1996)
24. Siegel, M.: HPSG Analysis of Japanese. In: Wahlster, W. (ed.) *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer (2000)
25. Rayner, M., Bouillon, P., Van Dalsem III, V., Isahara, H., Kanzaki, K., Hockey, B.A.: A Limited-Domain English to Japanese Medical Speech Translator Built Using REGULUS 2. In: Proceedings of 41st Annual Meeting of the Association for Computational Linguistics (interactive poster and demo track), Sapporo, Japan (2003)

Arabic Language Analyzer with Lemma Extraction and Rich Tagset*

Ahmed H. Aliwy

Institute of Informatics, University of Warsaw, Warsaw, Poland
aliwy@mimuw.edu.pl

Abstract. Arabic language analyzers have been studied constructed and studied by many researchers. Most of research projects and commercial applications in the area of Arabic Natural Language Processing (ANLP) build their own analyzer. Typically they are intended for just one project or application and can not be generalized to work in other areas of ANLP. All of these analyzers didn't cover our requirements from the analyzer which make us to build a new one. Our analyzer is also a part of a complete Arabic tagging system. It receives the output of a tokenizer in the form of inflected words, and produces for each of them a set of several possible analyzes, consisting of: a POS tag, features and a lemma. It differs from most of the existing analyzers because it produces a lemma rather than stem or root, which is a significantly harder task in Arabic, and because POS and features are described by a new very rich tagset, described in a previous publication. The test dataset was a small corpus of 16 k words, manually annotated by a single analysis for each word, correct for this particular use of that word. In the test, for 99.67% of words, the correct analysis was among those produced by the analyzer. On the other hand, in a manual verification of the output of the analyzer, only 0.1% of all analyses were grammatically incorrect.

Keywords: Arabic language analyzer, lemma generation, analyzer and POS tagging.

1 Introduction

The Arabic language is based on inflection and derivation, and words have many different forms that result from these procedures. Therefore extracting lemma is a hard problem for Arabic language. As a consequence, many researchers chose to deal with the stem, which is easier to extract, rather than with lemma. For example, in broken (abnormal) plural of nouns the word changes completely. In lemmatization the original form must be found, in stemming it is not necessary and is therefore easier.

Extracting lemma was much less studied than stem in the analysis stage. Many researchers dealt with lemma in Arabic language, but they did not explain details of the procedure of extracting lemma from the word. Some other researchers did not

* It is part of PhD dissertation of Whole Arabic Tagging System with rich tagset.

differentiate between lemma and stem and they dealt with them as if they were the same. The other researchers dealt with the root especially in morphological analyses. It should be noted that root induction is relatively simpler than stem and lemma.

In this paper we build Arabic analyzer which has two goals: the first is extracting POS and features of the word. The second is extracting the lemma of the word. These two goals are implemented in parallel. We built a dictionary as assistant for achieving these two goals.

The proposed analyzer is not intended for general use because it was designed and implemented as a preprocessing stage for Arabic Tagging system and, using the context of the word, it will reject some analyses, saving Tagger's work.

2 Lemma, Stem and Root

When we deal with analyzer, we must differentiate among three terms: Lemma, Stem and Root. They have different meaning. The lemma is the **canonical form, dictionary form, or citation form** of a set of words. A stem is the part of the word that never changes even when morphologically inflected¹. The root is the original letters of the word. Moreover, the term "root" is ambiguous in Arabic language where some researchers take it as letters, the others take it as the imperative verb in 3rd masculine.

When we deal with the root, then the derivational and inflectional morphology will be taken in account. When we deal with lemma, then only inflectional morphology will be taken in account. When we deal with stem, then part of inflectional morphology with part of derivational morphology will be taken in account. For example: changing the whole word will not be taken in account as broken plural. Figure 1 shows the difference among them with adding "number" feature to the word "كتاب" "kitAb"² "book".

Word	kitAb كتاب (book)	kitAbAn كتابان, kitAbYn كتابين (two books)	Kutub كُتُبُ (books)
Root	Ktb ك ت ب	Ktb ك ت ب	ktb ك ت ب
Stem	kitAb كتاب	kitAb كتاب	Kutub كتب
Lemma	kitAb كتاب	kitAb كتاب	kitAb كتاب

Fig. 1. Lemma, stem and root of the word "book" with adding number feature³

¹ In Arabic language the changing of vowels will be taken in account in stemming.

² We used buckwalter XML transliteration.

³ The plural is broken for this noun.

We can summarize the difference in the following points:

1. Stemming reduces word-forms to (pseudo) stems, whereas lemmatization reduces the word-forms to linguistically valid lemmas. Getting the root is done by reducing word-forms to original letters (root).
2. In case of root we deal with inflectional and derivational variants. In case of stem we deal with inflectional (partially) and derivational (partially) variants. In case of Lemma we deal with inflectional (completely) variants.
3. Extracting stem and root is relatively simple and can be done by deleting affixes. Extracting Lemma is more sophisticated and must refer to dictionary in some cases.
4. The root and stem are not valid words but lemma is.
5. More than one lemma can have the same stem; more than one stem can have the same root.

In our work, for verbal classes the lemma is 3rd masculine imperative verb. Lemma for the noun classes is the singular masculine, and if it does not exist, the singular feminine. For particles, strictly speaking, there is no lemma, so for unification we define it to be the particle itself.

3 Related Works

In [1], the authors do lemmatization in three phases: analyzing, POS tagging and then lemma generation. The first phase implementation is done with the open source Khoja stemmer [2] i.e. no private analyzer. The second phase is POS tagging which depends basically on Patterns. The third phase is lemma generation which is partially related to our approach. Our approach at the first glance may be appear similar this work but there are many differences: the important is that they take the output of POS tagging to Lemma generation and in our work the output of lemmatization and analyzing stage will be fed to POS tagging. I.e. our lemma generation is done by the analyzer alone and does not depend on tagging.

Concerning morphological analyzers, there are many works in this field.

[3] (Nizar-Rambow-Kiraz) MAGEAD provides an analysis for a root+pattern representation, it has separate phonological and orthographic representations, and it allows for combining morphemes from different dialects.

[4] Darwish analyzer was only concerned with generating the possible roots of a given Arabic word. It is based on automatically derived rules and statistics.

[5] (Gridach and Chenfour) their approach is based on Arabic morphological automaton technology. They take the special representation of Arabic morphology (root and scheme) to construct a few morphological automata which were used directly in developing a system for Arabic morphological analysis and generation.

[6] (Elixir-FM) is a functional morphology system which models templatic morphology and orthographic rules.

[7] (BAMA Buckwalter) is based on a Lexicon which has morphotactic and orthographic rules encoded inside it.

4 Challenges in Arabic Analysis

Due to the morphological complexity of the Arabic language, morphological analysis with lemma extraction is a very challenging task. Arabic language is regular in most cases of inflection and derivation, which leads to a relatively easy generation process. However, for irregular forms, it is more complicated. This difficulty grows rapidly also when a nonvowelized text is used⁴. Then the analysis process has to consider all possible vowelizations and produce all possible correct analyses for them. This huge number of analyses for each nonvowelized word leads to much increased probability of producing some wrong analyses among them.

The main challenges are:

1. A nonvowelized word can correspond to many vowelized words and therefore to many possible lemmas: for example the lemmas for the word “كتب” “ktb” can be “كُتِبَ” “kataba” (write), “كُتَابَ” “ktAb” (book) and “كُتِبَ” “kat~aba” (dedicated to write).
2. A normalized word can correspond to many unnormalized words and therefore to many possible lemmas: for example the lemmas for the word “ان” “An” can be “ان” “On” “ان” “In” “ان” “In” in unvowelized case.
3. Deleting or changing some letters, even in regular forms. For example the lemma for the word “يقول” “yqwl” (he say) is “قال” “qAl” (said).
4. Words whose grammatical lemma ends or begins with a sequence of letters identical to an affix. The mistake may occur when the attributes are extracted from the affixes. For example the letters “ون” could be falsely interpreted as a suffix and deleted from the word “مرهون” “mrhwn” (pawned). Similarly, the letters “ان” in the proper noun “عدنان” “EdnAn” could be interpreted as a suffix. Similarly, the letter “ت” in the common noun “تعاون” “tEAwn” “cooperation” could be interpreted as a prefix.
5. Complete change of the word in regular and irregular cases: broken plural is often an example of this phenomenon. The best solution in this case is to use a dictionary.
6. Transliterations of foreign words. Many foreign words, for instance foreign proper nouns, have more than one form of Arabic transliteration, which affects the analyzing process.

In our complete system clitics are dealt with during tokenization stage, and hence are not listed here.

5 Analyzing as Tagging Preprocessing

Arabic analyzing is the second preprocessing step, after Tokenization step, of the whole tagging system which we propose. Therefore we suppose that the input word to

⁴ Traditionally Qur’ān is vowelized, and so are children’s books. The rest of present day texts are nonvowelized.

analyzing is an inflected word or clitics as in figure 2. The output of this stage will be lemma, POS and features in case of nouns and verbs, meaning and working in case of particles⁵.

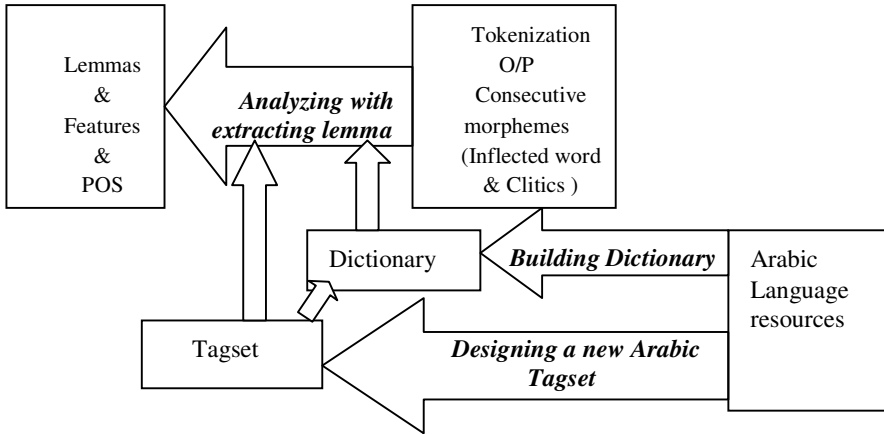


Fig. 2. Analyzing and extracting lemma as tagging preprocessing

Most of researchers depend basically on patterns for extracting root or stem but the pattern, in most cases, is not an efficient way for extracting lemma from the word. There is no any standardization for producing lemma from the word form in most cases.

We used our own lemmatizer and analyzer and did not use existing morphological analyzers for the following reasons:

1. We proposed a new Arabic Tagset and existing morphological analysers will not extract all the POSs and the features consistent with this tagset.
2. We deal with lemma instead of the root and stem.
3. We want to implement a complete tagging system.
4. Most analyzers mix segmentation and analyzing in one stage but we separate them into different tasks.

6 Our Analyzing Approach

We suppose that the input to analyzing process is a word without clitics (inflected word alone)⁶ or clitics alone. However, we assume that we are processing text and therefore the context of the present word is known to the analyzer.

⁵ Our Tagset consists of 5 main classes which are Noun, Verb, Particle, Residual and punctuation. Noun main class has 14 subclasses and 4 features (as gender, number ...), Verb main class has 3 subclasses and 7 features. Particals has 7 subclasses (particle working) and 21 features (particle meaning). See figure 3 for sample of the used tags.

⁶ Inflected word is any word without clitics.

The Analyzer deal with known and unknown words in different ways.

1. Known words processing: no processing is needed because the lemma and features are in the dictionary.
2. Unknown words processing: we have more things to do. Unknown words are more likely to be nouns, because we use a large and fairly complete database of inflected verbs in the dictionary. As we mentioned previously there are many classes of nouns which are closed sets (like e.g. relative nouns). The open classes of nouns are: proper, common, adjectives including genealogical and reduced nouns⁷.

We will explain the construction of the dictionary in the next section. Now, we will focus on processing the unknown words.

6.1 Unknown Words Processing

Our approach to processing unknown words is to do the most likely analyzing, without exhausting all possibilities. The main steps for unknown words processing are:

1. Extracting POS possibilities.
2. Extracting lemma and features.

Extracting the POS Possibilities. First we need to know the main POS to the word (noun, verb or particle); however, particle can be eliminated because the particles form a closed set. So really we have only two possibilities: noun and verb. Then we will extract the POSs according to our Tagset.

1. **Extracting the Main POS:** We must decide: verb or noun in this step. It can be done by applying the following classes of rules:
 - (a) Clitics rules. As example the definition particle “ل” “Al” (the) appears with noun only.
 - (b) Affixes and word structure rules. For example, the letter “س” “p” appears in nouns only.
 - (c) Context rules. For example: verb cannot follow another verb.
 - (d) If none of the above rules is applicable, we assume by default that the word is a noun.
2. **Extracting the POSs According to Our Tagset:** It can be done separately for verbs and noun subclasses:
 - (a) Identification of past, present and imperative forms of verbs is achieved by:
 - (i) Proclitics: for example if the word has proclitic “س” “s” (will) it must be present tense verb.
 - (ii) Affixes: for example if the word has one of the prefixes “ي y, ا A, ن n, ت t”, it seems to be present tense verb.
 - (iii) Preceding word: for example, the word after “لن” “In” (not) must be a verb in present tense.

⁷ Our Tagset has 14 subclasses of noun.

- (b) Induction of Noun subclasses: proper, common, reduced and adjective (including genealogical)⁸ nouns is achieved by:
- (i) By the pattern: for example reduced nouns can be identified by their pattern because there are exactly three patterns for reduced nouns.
 - (ii) By the word structure: for example the genealogical can be induced by the word ending, because it always ends with “ي” “y”.
 - (iii) By the affixes: for example if the word ends with “ة”, it seems not to be a proper noun.
 - (iv) By the context: for example, if the previous word is a verb then the current one seems to be a common or proper noun.

At this stage we do not solve the ambiguities; instead we find the most important analyzing for the word. We may overlook some possibilities, but they are very infrequent.

Extracting Lemma and Features. After differentiation between classes of words now we do the second phase of extracting lemma and features. Verbs and noun subclasses will be processed separately, but by the same methodology:

1. Extracting the features from the affixes.
2. Extracting the lemma by:
 - (a) Deleting the affixes.
 - (b) Retrieving the deleted and (or) the changed letter which resulted from the inflection.

For verbs classes, the above steps will be:

1. From affixes: for example the verb ending with “ين” “yn” seems to be (i) masculine plural or (ii) singular feminine or (iii) dual masculine or (iv) dual feminine. If a verb starts with “ت” “t” it seems to be (i) masculine 2nd person or (ii) feminine 2nd person. If we combine these two rules on the verb “تقولين” “tqwlyn” (you (feminine) say), we simply induce its features to be singular feminine 2nd person.
2.
 - (a) Deleting the affixes. “تقولين” “tqwlyn” will give “قول” “qwly”
 - (b) “قول” “qwly” will become “قال” “qAl” (he say). Let us note this affects only the vowels “و, ا, ي”.

An example for nouns:

1. From affixes: for example a word ending with “ة” “p” seems to be singular feminine.

⁸ We take only these classes of nouns because other nouns subclasses are closed.

2.

- (a) Deleting affixes (with exceptions). For example the word “فتاة” “ftAp” (girl) will become “فتا” “ftA”. The word “جرثومي” “jrvwmy” (Bacterial) is a genealogical noun⁹ and, by exception, the affix “ي” will not be deleted.
- (b) Extracting the lemma by retrieving the deleted or changed letters (if necessary). “فتا” “ftA” will become “فتى” “ftY” “boy”.

We must remember that in most cases the word exists in the dictionary, which is quite large, especially for verbs, and the above heuristic analysis is done only for words which are not in the database.

7 Building Dictionary

Now we describe the construction of dictionaries, which are used in preprocessing. These dictionaries play a similar role to the dictionaries used in Buckwalter Analyzer, with lemma added to POS and Features.

For Verbs: This dictionary consists of little more than 6000 verbs inflected in all possible forms according to the templates used by Al-Dahdah[8] with adding certainty and jussive case. Then all these inflections are sorted and encoded in a way such that we can find them efficiently. The input to dictionary is an inflected verb in any tense or case and the output are its lemma and features. We used this large dictionary for one reason which is to get rid the problem of the changing which happened in the inflected verb. The second reason is that the verbs seem to be an almost closed set, and using about 6000 inflected verbs will give us more information than a corpus having 10 Mega words. The reason is that each verb has approximately 164 inflections. It means that we have approximately 984000 inflections, many of which will be missing in a corpus of size 10 Mega words.

At present the software does binary search in a full dictionary of about 984000 inflections and is reasonably fast.

However, it is possible to encode the dictionary in a smaller and slightly more effective data structure, which has separate dictionaries of prefixes, suffixes and pairs (stem*, lemma), similarly as in Buckwalter analyzer. Stem* is created exactly as a stem, but in some cases can be an illegal word, and therefore not a stem in the strict sense. Our task is to induce the possible stems* from the inflected form of the verb. For example when the verb “قال” “qAl” (he say) is inflected, we get as possible stems*: “قال” “qAl”, “قول” “qwI”, “قيل” “qyI” and “قل” “qI”, all of which point to the same lemma, which is the output. Indeed, only the first one of the above words is legal and is therefore a true stem. In other words, the stored stems* of inflected verbs are the forms which appear at least once in an inflection of a verb.

In case of particles we have a list of all particles, each one with its working and meaning, and therefore the analyzing process is again a simple search problem, like in the in cases of verbs.

⁹ The ambiguity between “كتابي” as my book or genealogical noun was solved by tokenization preprocessing.

In the case of nouns, adjectives and so on, we collected them from the Internet. We added inflections and derivations as feminine (if applicable), numbers, genealogically (Yaa Alnasabi) and reduced nouns. The object, subject nouns, broken plural and so on are not derived by this method; instead they are collected from texts which reduce the cost of the code (time of writing code) and applying this generation on them if applicable. There are many classes of nouns which are closed sets, for example question nouns, numeral nouns and so on. The resulting dictionary is updatable.

8 Results

The proposed analyzer was built as a preprocessing stage of an Arabic Tagging system. It is therefore not a general purpose analyzer. It produces all possible analyses for a given inflected word or clitics. These analyses are POS, features and lemma. Because it is used for subsequent tagging, the evaluation of it should measure how well it satisfies its function, i.e., generates true combinations of Tag (POS & features) and lemma. Therefore we will not evaluate it according to Recall, Precision and F-measure.

The first important thing is to have the true Tag and lemma produced.

The test dataset was a small corpus of 16 k words, manually annotated by a single analysis for each word, correct for this particular use of that word. In the test, for 99.67% of words, this correct analysis was among those produced by the analyzer.

The second important thing is that the analyzer never produces grammatically incorrect analyses.

In a manual verification of the output of the analyzer, only 0.1% of all analyses were grammatically incorrect. Figure 3 shows practical analyzing for simple sentence.

9 Discussion and Feature Work

We have built, implemented and evaluated an Arabic analyzer which extracts lemma. The analyzer produces POS, features and lemma of the inflected word or clitics. The produced POS and feature are described according to our new, very rich Tagset. Many problems, which can be solved in Tagging system, were solved by the analyzer using the context. The context is taken in account only for unknown words. According to the previous results, it is suitable to use it in Tagging. Lemma extraction offers many benefits when compared to extracting stem or root. For example, it can be used in word sense disambiguation.

Our suggestion is that expanding (i) the number of the inflected verbs used in the analyzer and (ii) expanding the database of abnormal inflections of the noun subclasses, can lead to still more accurate analyses.

It would be very beneficial to test the analyzer on a larger corpus. However, it is very time-consuming to produce, since it must be done by hand, since we use a new, rich tagset.

Word		Transliteration	Tokens	Lemma	Analyzing / Tag	Meaning of Analyses(Tag meaning)
Arabic	Transliteration					
اجتمع	AjmE	Met	اجتمع	اجتمع	VPst_3SMOYNA	Verb Past Third Singular Masculine NonMoodStructured Not Certainty Active
				اجتمع	VImv_2SMOYNA	Verb Imperative Second Singular Masculine NonMoodStructured Not Certainty Active
				اجتمع	VPrt_1SMNNA	Verb Present First Singular Masculine Nominative Not Structured Not Certainty Active
				اجتمع	VPrt_1SMJNNA	Verb Present First Singular Masculine JussiveNot Structured Not Certainty Active
				اجتمع	VPrt_1SMANNA	Verb Present First Singular Masculine Accusative Not Structured Not Certainty Active
				اجتمع	VPst_3SMOYNP	Verb Past Third Singular Masculine NonMoodStructured Not Certainty Passive
				اجتمع	VPrt_1SMNNP	Verb Present First Singular Masculine Nominative Not Structured Not Certainty Passive
				اجتمع	VPrt_1SMJNPP	Verb Present First Singular Masculine JussiveNot Structured Not Certainty Passive
زعماء	zEmA'	Leaders	زعماء	زعيم	NAdo_PMNN	Noun Adjective(Other) Plural Masculine Nominative Not Structured
				زعيم	NAdo_PMAN	Noun Adjective(Other) Plural Masculine Accusative Not Structured
				زعيم	NAdo_PMGN	Noun Adjective(Other) Plural Masculine Genative Not Structured
الدول	Aldwl	States	الدول	ال	PNon_Def	Particle Not_have_working_have_meaning_of_Definition
				دولة	NNou_PFNN	Noun Common Plural Feminine Nominative Not Structured
				دولة	NNou_PFAN	Noun Common Plural Feminine Accusative Not Structured
				دولة	NNou_PFGN	Noun Common Plural Feminine Genative Not Structured
العربية	AlEr-by	The Arabic	عربية	ال	PNon_Def	Particle Not_have_working_have_meaning_of_Definition
				عربية	NAdg_SFNN	Noun Adjective(Genealogical) Singular Feminine Nominative Not Structured
				عربية	NAdg_SFAN	Noun Adjective(Genealogical) Singular Feminine Accusative Not Structured
في	fy	In	في	في	PRed_Non	Particle For_Reduction have_No_meaning
				في	PRed_Cau	Particle For_Reduction have_meaning_of_Caution
				في	PRed_Adv	Particle For_Reduction have_meaning_of_Adverbial
				في	NFiv_SMGN	Noun Five_Noun Singular Masculine Genative Not Structured
				وفي	VImv_2SFOYNA	Verb Imperative Second Singular Feminine NonMoodStructured Not Certainty Active
بغداد	bgdAd	Baghdad	بغداد	بغداد	NPrp_SMNN	Noun Proper Singular Masculine Nominative Not Structured
				بغداد	NPrp_SMAN	Noun Proper Singular Masculine Accusative Not Structured
				بغداد	NPrp_SMGN	Noun Proper Singular Masculine Genative Not Structured
				بغداد	NNou_SMNN	Noun Common Singular Masculine Nominative Not Structured
				بغداد	NNou_SMAN	Noun Common Singular Masculine Accusative Not Structured
				بغداد	NNou_SMGN	Noun Common Singular Masculine Genative Not Structured
والجمعة وا	wA-jmEwA	Andgather	و	و	PCnj_Lnk	Particle For_Conjection have_meaning_of_Linking
			واجمعوا	و	PRed_Cer	Particle For_Reduction have_meaning_of_Certainty

Fig. 3. Practical analyses for simple sentence

				و	PRed_Non	Particle For_Reduction have_No_meaning
				أجمعع	VPst_3PMOYNA	Verb Past Third Plural Masculine NonMoodStructured Not Certainty Active
				أجمعع	VImv_2PMOYNA	Verb Imperative Second Plural Masculine NonMoodStructured Not Certainty Active
				أجمعع	VPst_3PMOYNP	Verb Past Third Plural Masculine NonMoodStructured Not Certainty Passive
				جمعع	VImv_2PMOYNA	Verb Imperative Second Plural Masculine NonMoodStructured Not Certainty Active
على	Eiy	To/on	على	على	PRed_Lnk	Particle For_Reduction have_meaning_of Linking
				على	PRed_Non	Particle For_Reduction have_No_meaning
				على	PRed_Adv	Particle For_Reduction have_meaning_of Adverbial
				على	PRed_Cnd	Particle For_Reduction have_meaning_of Conditional
				على	PRed_Cau	Particle For_Reduction have_meaning_of Caution
				على	VPst_3SMOYNA	Verb Past Third Singular Masculine NonMoodStructured Not Certainty Active
				على	PCop_Cer	Particle For_coppying have_meaning_of Certainty
ان	An	That	ان	ان	PCop_Cer	Particle For_coppying have_meaning_of Certainty
				ان	PNon_Non	Particle Not_have_working have_No_meaning
				ان	PNon_Non	Particle Not_have_working have_No_meaning
				ان	PNon_Neg	Particle Not_have_working have_meaning_of Negative
				ان	PNon_Non	Particle Not_have_working have_No_meaning
				ان	PNon_Non	Particle Not_have_working have_No_meaning
				ان	PAcu_Sub	Particle For_Accusative have_meaning_of Subordinating
				ان	VPst_3SMOYNA	Verb Past Third Singular Masculine NonMoodStructured Not Certainty Active
				ان	VPst_3SMOYNP	Verb Past Third Singular Masculine NonMoodStructured Not Certainty Passive
				ان	VPrt_1SMJNNA	Verb Present First Singular Masculine JussiveNot Structured Not Certainty Active
يساندوا	ysAndwa	They support	يساندوا	سندت	VPrt_3PMJNNA	Verb Present Third Plural Masculine JussiveNot Structured Not Certainty Active
				سندت	VPrt_3PMANNA	Verb Present Third Plural Masculine Accusative Not Structured Not Certainty Active
				سندت	VPrt_3PMINNP	Verb Present Third Plural Masculine JussiveNot Structured Not Certainty Passive
				سندت	VPrt_3PMANNP	Verb Present Third Plural Masculine Accusative Not Structured Not Certainty Passive
الربيع	AlrbyE	The spring	ربيع	ال	PNon_Def	Particle Not_have_working have_meaning_of Definition
				ربيع	NPrp_SMNN	Noun Proper Singular Masculine Nominative Not Structured
				ربيع	NPrp_SMAN	Noun Proper Singular Masculine Accusative Not Structured
				ربيع	NPrp_SMGN	Noun Proper Singular Masculine Genitive Not Structured
				ربيع	NPrp_SMNN	Noun Proper Singular Masculine Nominative Not Structured
				ربيع	NPrp_SMAN	Noun Proper Singular Masculine Accusative Not Structured
				ربيع	NPrp_SMGN	Noun Proper Singular Masculine Genitive Not Structured
				ربيع	NNou_SMNN	Noun Common Singular Masculine Nominative Not Structured
				ربيع	NNou_SMAN	Noun Common Singular Masculine Accusative Not Structured
				ربيع	NNou_SMGN	Noun Common Singular Masculine Genitive Not Structured
العربي	AlErby	The Arabic	عربي	ال	PNon_Def	Particle Not_have_working have_meaning_of Definition
				عربي	NPrp_SMNN	Noun Proper Singular Masculine Nominative Not Structured
				عربي	NPrp_SMAN	Noun Proper Singular Masculine Accusative Not Structured
				عربي	NPrp_SMGN	Noun Proper Singular Masculine Genitive Not Structured
				عربي	NAdg_SMNN	Noun Adjective(Genealogical) Singular Masculine Nominative Not Structured
				عربي	NAdg_SMAN	Noun Adjective(Genealogical) Singular Masculine Accusative Not Structured
				عربي	NAdg_SMGN	Noun Adjective(Genealogical) Singular Masculine Genitive Not Structured

Fig. 3. (continued)

			تَرْبِي	NAdg_SMNN	Noun Adjective(Genealogical) Singular Masculine Nominative Not Structured
			تَرْبِي	NAdg_SMAN	Noun Adjective(Genealogical) Singular Masculine Accusative Not Structured
			تَرْبِي	NAdg_SMGN	Noun Adjective(Genealogical) Singular Masculine Genitive Not Structured
			تَرْبِي	NAdg_SMNN	Noun Adjective(Genealogical) Singular Masculine Nominative Not Structured
			تَرْبِي	NAdg_SMAN	Noun Adjective(Genealogical) Singular Masculine Accusative Not Structured
			تَرْبِي	NAdg_SMGN	Noun Adjective(Genealogical) Singular Masculine Genitive Not Structured
			تَرْبِي	NAdg_SMNN	Noun Adjective(Genealogical) Singular Masculine Nominative Not Structured
			تَرْبِي	NAdg_SMAN	Noun Adjective(Genealogical) Singular Masculine Accusative Not Structured
			تَرْبِي	NAdg_SMGN	Noun Adjective(Genealogical) Singular Masculine Genitive Not Structured
			تَرْبِي	VImv_2SFOYNA	Verb Imperative Second Singular Feminine NonMoodStructured Not Certainty Active

Fig. 3. (continued)

References

1. Tarek, E., Fatma, E.: An Accurate Arabic Root-Based Lemmatizer for Information Retrieval Purposes. IJCSI International Journal of Computer Science Issues 9(1,3) (January 2012)
2. Khoja, S.: Stemming Arabic Text. Computing Department, Lancaster University, Lancaster (1999)
3. Nizar, H., Owen, R., George, K.: Morphological Analysis and Generation for Arabic Dialects. In: Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, Michigan, USA (2005)
4. Kareem, D.: Building a Shallow Arabic Morphological Analyzer in One Day. In: Proceedings of the ACL 2002 Workshop on Computational Approaches to Semitic Languages, PA, USA (2002)
5. Mourad, G., Noureddine, C.: Developing a New System for Arabic Morphological Analysis and Generation. In: Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011, Chiang Mai, Thailand, pp. 52–57 (2011)
6. Otakar, S.: Functional Arabic Morphology, Formal System and Implementation. Doctoral Thesis, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Prague (2007)
7. Buckwalter, T.: Issues in Arabic Orthography and morphology Analysis. In: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, COLING, Geneva (2004)
8. El-Dahdah, A.: An Intermediate Dictionary of Verb Conjugation, 1st edn. Libaririe Du Liban, Beirut (1994) (in Aabic)

Tracking Researcher Mobility on the Web Using Snippet Semantic Analysis

Jorge J. García Flores¹, Pierre Zweigenbaum¹, Zhao Yue²,
and William Turner¹

¹ LIMSI - CNRS, B.P. 133, F-91403, Orsay Cedex, France

² Université Paul Valéry, Route de Mende, F-34199, Montpellier, France
{jgflores,pz,turner}@limsi.fr, yue.zhao@etu.univ-montp3.fr

Abstract. This paper presents the *Unoporuno* system: an application of natural language processing methods to the sociology of migration. Our approach extracts names of people from a scientific publications database, refines Web search queries using bibliographical data and decides of the international mobility category of a person according to the location analysis of those snippets classified as mobility traces. In order to identify mobility traces, snippets are filtered with a name validation grammar, analyzed with mobility related semantic features and classified with a support vector machine. This classification method is completed by a semi-automatic one, where *Unoporuno* selects 5 snippets to help a sociologist decide upon the mobility status of authors. Empirical evidence for the automatic person classification task suggest that *Unoporuno* classified 78% of the mobile persons in the right mobility category, with $F=0.71$. We also present empirical evidence for the semi-automatic task: in 80% of the cases sociologist are able to choose the right category with a moderate level of inter-rater agreement (0.60) based on the 5 snippet selection.

1 Introduction

Among the Latin-American authors who published scientific articles about biotechnology during 2011, how many of them are living abroad? And how many of them have studied in foreign universities before coming back to work in their home countries? Sociologists of migration, and in particular those working on the “brain drain” issue—that is, the idea that talent mobility is a serious problem affecting developing countries—find it hard to answer such fine-grained questions using traditional data sources such as demographic registers, labour surveys or population census, which require a great deal of field work and are carried out too infrequently to provide a constantly updated picture of talent mobility [1]. They have been experimenting other methods such as using browsers to search the Web for biographical evidence of mobility but are faced with the “needle in the haystack” problem [2]: it takes them a great deal of time to wade through the results of a browser search (hundreds of snippets) to find the precise evidence they need (a CV, a personal Web Page, etc.) to classify a person in one of the following categories:

- Mobile: has gone abroad for professional or academic reasons and has lived away from the country of origin for at least one year.
- Local: has only spent short periods of time abroad (less than one year).

Web People Search (WePS) systems [3] are concerned with clustering the results of ambiguous name queries in order to distinguish between people with the same names. Our system also aims at finding people on the Web, however, it differs from WePS because its starting point is not a user query, but a publication record. Information can consequently be extracted on, for example, an author’s geographical location, his or her affiliation or topics from the publication’s title in order to refine name-only queries. We call this the *Mobility Traces Classification* (MTC) task. This paper presents *Unoporuno*: an NLP system for carrying out the MTC task. Its main contribution consists in implementing a metasearch engine based upon bibliographical query refinement and multilingual Web search. The resulting snippets are first filtered using a personal name grammar that recognizes valid name variations; then classified on the basis of the mobility-related features they contain; and finally ranked statistically according to their calculated relevance for deciding on the mobility status of a person. We present two variants of the MTC task: an automatic one, where *Unoporuno* decides on the mobility status of a person based on a location analysis of the top ranked snippets, and a semi-automatic one, where only the “top five” snippets in the ranking are presented to a sociologist, who manually attributes a mobility status. The article is organized as follows: Section 2 presents related work; Section 3 provides methodological details about the *Unoporuno* pipeline; Section 4 presents an overall evaluation of each step of the pipeline; Section 5 presents the results, and Section 6 discusses these results and outlines future work.

2 Related Work

Evaluation campaigns of the Web People Search task [4, 5, 6, 3] generally focus on ways of clustering documents into sets which characterize different persons that share the same name. Quoted-named queries are used as input to the WePS task, but with no query refinement. Ariles et. al. [7] showed that when query refinement is used, in most cases relevant pages are found, but they note that human users seldom know what refinement terms to use in order to produce these positive results. In contrast, our MTC task provides a semantically rich context for Web People Search. Its input is a bibliographical record, from which we extract topics, organizations and locations to enrich multilingual name queries. However, this greatly increases the number of Web queries and search results (an average of 400 snippets per person, compared to 100 for WePS). For that reason, instead of directly processing Web pages, *Unoporuno* implements a common strategy [8] which consists in filtering and classifying the snippets found by the search engine. This requires extracting suitable features directly from these snippets, and implementing a statistical classification of Google snippets based upon the semantic features of mobility. Previous work on the linguistics and semantics of Web search has largely focused on queries [9, 10], but relatively few

studies have focused on snippets, even though an eye tracking study has shown that snippets are looked at longer [11] than titles, images or URL address.

3 NLP Pipeline

*Unoporuno*¹ is a metasearch engine for query refinement and snippet classification (see Figure 1). Its input is a Web of Science (WoS) data extraction: e.g., all the biotechnology publications of Uruguayan researchers in October 2011. The output consists of 5 Web search snippets that are presented to the sociologist in order to classify the person in one of the above mentioned categories. This section describes each of the steps of the *Unoporuno* processing pipeline.

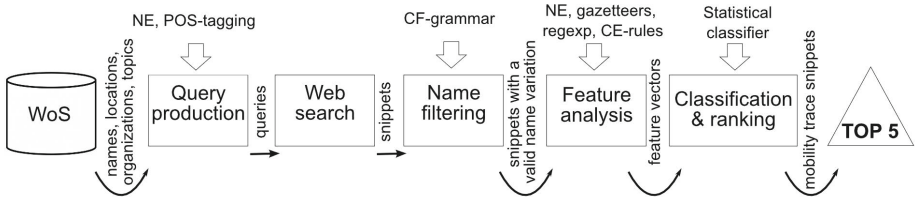


Fig. 1. NLP pipeline for Mobility Traces Classification

3.1 Pre-processing and Query Production

The pre-processor extracts author names, publication titles, organizations and locations from the input file, which is a bibliographical extraction from the ISI Web of Knowledge². The ISI export format separates author, publication title and affiliation in different columns. The first step of the pre-processing consists in extracting geographical locations and organizations from the author's affiliation. Authors are then filtered by affiliation and name. Researchers affiliated to non-Spanish speaking countries are filtered out, except for those with a Spanish first or last name. A Spanish name list was built from census³ data and geo-demographic analysis for that purpose [12]. The second step focuses on query refinement and production. Names of people are combined with noun phrases from the publication's title; these noun phrases are identified using the Freeling bilingual POS-tagger and NE recognizer [13]. Geographical locations are translated into Spanish or English and multilingual queries are generated. When an organization's language is neither of the two (for instance, the *Karolinska Institutet* from Sweden) queries are built using Spanish, English and the organization's language (Swedish in our example). Language detection is made using

¹ Open source available at <https://github.com/unoporuno/unoporuno>

² <http://www.isiwebofknowledge.com/>

³ <http://www.ine.es/daco/daco42/nombyapel/nombyapel.htm>

Google translator; city and country translation with home-made gazetteers. An average of 19 queries per person are produced⁴.

3.2 Name Matching Filter

The mass of snippets resulting from Web search queries is then filtered to select those with a valid variation of the person's name. A context-free grammar for Spanish and English names was developed for this purpose⁵ (see Table 1). By successive recursive operations on the parse tree of the name, we produce a set of regular expressions which recognize name abbreviations, initial expansions, first and last name inversions or partial name suppression in the snippet. For the test set (see Table 4), the name filtering process retained 3,476 snippets among the 11,266 retrieved by the queries. The grammar has four terminal elements: 1) a common name, 2) a name with a particle (*Ana Ozores de Clarin*), 3) a name with a typographical link (*Ana Ozores-Clarin*) and 4) the initial. The syntax requires at least a name (or an initial) and a last name. It takes into account Spanish name formation rules, using father and mother last names, and ensures that one of the last names cannot be compressed as an initial. Table 2 describes valid operations on any branch of the Spanish name grammar tree. Operations are implemented as recursive algorithms that compress, expand or suppress elements from the tree. Regular expressions are produced and then used to validate name variations found in the snippets. When any given regular expression is true, the snippet is considered as containing a valid name variation. A total of 27 possible variations for a name were identified during the acquisition process.

Table 1. Context-free grammar for Spanish and English name parsing

PersonName	→	FirstName LastName
FirstName	→	FirstName Initial
FirstName	→	FirstName CommonName
FirstName	→	Initial
FirstName	→	CommonName
FirstName	→	TypoName
FirstName	→	CommonName ParticleName
LastName	→	MainLastName
LastName	→	MainLastName CommonName
LastName	→	MainLastName ParticleName
LastName	→	MainLastName TypoName
LastName	→	MainLastName Initial
MainLastName	→	CommonName
MainLastName	→	ParticleName
MainLastName	→	TypoName

⁴ Google querying using P. Krumins Python Library, <http://bit.ly/EUizu>

⁵ It was implemented with the NLTK formal grammar library [14].

Table 2. Regular expressions generated from the name grammar to check valid name variations

Operation	Description
<i>n</i> : name	$n \rightarrow N$
<i>a</i> : surname	$a \rightarrow A$
<i>C</i> : compression	$CnLa(\text{Noe Lopez}) \rightarrow N \setminus .? \text{Lopez}$
<i>E</i> : expansion	$EnLa(\text{Eva M Perez}) \rightarrow \text{Eva M}[a - z]^+ \text{Perez}$
<i>L</i> : literal	$LnLa(\text{Noe Lopez}) \rightarrow \text{Noe Lopez}$
<i>X</i> : extra element	$LnXLa(\text{Eva M Perez}) \rightarrow \text{Eva M}[A - Z][a - z]^+ \text{Perez}$
<i>V</i> : inversion	$VCnLa(\text{Eva M Perez}) \rightarrow \text{Perez},? +E[\setminus,]?[-]?M \setminus .?$
<i>SI</i> :suppress initial	$SInSIa(\text{Noe J Lopez F}) \rightarrow \text{Noe Lopez}$

3.3 Semantic Features Analysis

Feature analysis consists in searching in the snippet content for mobility related information. The rationale is that the snippet contents might give clues about mobility traces not directly visible in the snippet, but which are contained in the referred to document. Feature analysis is performed by means of regular expression and gazetteers. To design the multilingual rules, an extensive n-gram based analysis of the 58,220 snippets from the training set (see Section 4.1) and NE's from the JRC base [15] was performed. Acronyms received special treatment: a list of uppercase sequences were extracted from all snippets of the test set and transformed into content-specific *Unoporuno* queries whose results were then analyzed to find significant acronyms for mobility. Most of the features are binary. The underlying idea is to represent a snippet as a vector of binary features. Table 3 shows semantic features used to analyze snippets. Features 1 to 8 convey very simple information, while features 9 to 14 capture more complex phrases (biography, profession, academic background) that needed a deeper linguistic analysis. Regular expressions and gazetteer rules were preferred to deeper linguistic techniques because of the multilingual character of snippets.

3.4 Snippet Classification and Ranking

The last step statistically classifies and ranks the snippets. The ranking is used both by the automatic Mobility Traces Classification (MTC) task (to attribute a mobility status to a person) and by the semi-automatic MTC task (to select the top-5 snippets that will be presented to the sociologist). Four classifiers from the Weka toolkit (Decision trees, Naive Bayes, NBTrees and SVM) were trained on the training set and tested on the test set snippets (see Table 4). The classification process takes all the snippets of a person, classifies them and then ranks those classified as mobile to select the top-5. Classifiers were trained on three categories: strong mobility trace, weak mobility trace and no trace. Mobility traces are considered strong if both points of the movement (origin and destination) are visible in the document referred to the snippet. Traces are considered as weak if only one point of the potential movement is visible. We use

Table 3. Semantic features for snippet analysis (*cities have more than 100,000 hab)

Name	Type	Description	Type
PhD thesis	regex	The snippet links to a PhD thesis	bool
LinkedIn	gazet	The snippet links to a LinkedIn Web page	bool
Publication	gazet	The snippet links to a scientific publication	bool
e-mail	regex	The snippet contains an email	bool
Non Latin-American nationality	gazet	The snippet contains a nationality from a non Latin-American country	bool
Latin-American nationality	gazet	The snippet contains a nationality from a Latin-American country	bool
Person name found in <i>URL</i>	regex	Personal first or last name in the <i>http</i> address	bool
CV	regex	The snippets links to a CV	bool
Profession	regex	The snippet contains a profession name	bool
Degree	regex	The snippet contains academic information	bool
Biographical sentence	regex	The snippet contains a biographical sentence	bool
Organization acronym	gazet	The snippet contains an organization acronym	bool
City & region	gazet	The snippet contains a city or region name	bool
Country	gazet	The snippet contains a country name	bool
Organization	regex	The snippet contains an organization name	bool
Feature count	-	Number of features found in the snippet	int

a geographical heuristic to select the top-5 mobility snippets. First, we extract all the geographical locations from snippets classified as strong and weak mobility traces. Second, we calculate frequent countries from those locations. Finally, we include in the top-5 three snippets containing locations outside Latin-America, and two containing Latin-American locations. If one of both locations is missing, snippets are sorted in decreasing order of their feature count.

3.5 Person Classification

In the automatic MTC task, persons are classified using geographical data found in the snippets, with no sociologist annotation at all. First, the title and the description of those snippets classified as mobility traces are parsed to extract locations. Second, locations are associated to a country. For this experiment, we consider only cities and countries as locations. Neither organizations nor nationalities nor any element of the URL are associated to a country yet. The relation between a city and a country is obtained through a qualified gazetteer of 3,545 world cities with more than 100,000 inhabitants extracted from Wikipedia. Finally, the person is classified according to the most frequent countries in the snippet selection: if Latin-American and non Latin-American countries are found in the frequent countries list, the person is classified as mobile; otherwise, the person is classified as local. If no locations are found in the snippet list, the person is not classified.

4 Evaluation

4.1 Data

Table 4 summarises the data collected for mobility trace classification. Training and test datasets have no overlap. The training set comes from two sources. First, 102 researchers from Argentina, Colombia and Uruguay extracted from WoS, and whose mobility traces were annotated manually by sociologists. Second, 646 Latin-American researchers from WoS who were treated by *Unoporuno* using the baseline top-5 classification criteria. For each of these 646 researchers, the top-5 snippets were manually annotated. The test set was created using information collected from an on-line survey of Uruguayan researchers: 25 of these researchers answered that they live abroad or have been abroad for longer than a year for professional or academic purposes (mobile category). The other 25 answered that they had only spent short periods abroad (local category). Each document that a snippet pointed to was manually annotated as:

Table 4. Two hand annotated corpora for the MTC task.

Gold standard	Training set	Testing set
Researchers	102+646	50
Home country	Argentina, Uruguay, Colombia	Uruguay
Queries	782+10471	609
Filtered snippets	5544+52676	3476
Mobility traces	397+214	134
Home/destination country traces	921+770	1091 (home) 252 (dest)
No trace	4226+51692	1999

- Mobility trace: the snippet links to a document containing clear evidence of international mobility.
- Destination country trace: the snippet links to a document containing partial evidence of mobility (e.g., an affiliation to a foreign university).
- Home country trace: the linked document shows no international mobility, but an affiliation of the researcher to his home country.
- No trace: none of the above.

4.2 Name Matching Filter Evaluation

As written above, the name matching filter selects those snippets containing valid variations of a person name (see Section 3.2). To evaluate the grammar and regular expressions used for this step, we first selected on a random basis 100 snippets containing a valid variation of 10 persons names (positives) and 100 snippets containing no valid variation of 10 person names (negatives). Then we manually annotated false positives from the first set and false negatives from the second.

4.3 Semantic Features Evaluation

Two tests were used for semantic feature evaluation. First, a detailed evaluation of individual feature performance, and second, an evaluation of the impact of each feature on the whole automatic MTC task. Then, we performed ablation tests of the automatic MTC task in order to evaluate the impact of each feature on the main task. For each of the 15 features, we made a random selection of 50 snippets with the feature on (positives) and 50 snippet with the feature off (negatives). Then we annotated false positives from the first set and false negatives from the second. For the feature impact evaluation on the overall task, we trained 15 SVM ablated classifiers. An ablated classifier is trained by removing one feature from the original 15 feature set. The automatic person classification process was run 15 times with a 14 feature-set classifier, and the results compared to the full 15 feature-set run.

4.4 Snippet Classifiers Evaluation

We selected the best classifier by evaluating the top-5 snippets in two ways. First, we measured P@5 (precision at the fifth snippet), R@5 (recall at the fifth snippet) and F@5 based on the observed category value of each snippet. Second, we simulated whether a sociologist would be able to make a decision based on the top-5 snippets. This would be the case if at least one snippet allowed the sociologist to classify a person in the right category. Table 5 shows how to decide whether a snippet has this property given how it was manually annotated and the person’s true mobility status. Based on this we defined the Oracle Decision Rate (ODR) of a classifier as the proportion of persons for which the top-5 has this property. We also computed that rate for the mobile persons only (mODR).

4.5 User Evaluation of the Semi-automatic MTC Task

We evaluated the ability of sociologists to classify persons given the top-5 snippets produced by the classifier that obtained the best ODR. Three pairs of sociologists classified subsets of 10 persons (5 mobile, 5 local) of the test set. A seventh sociologist was asked to classify the entire test set (50 persons: 25 mobile, 25 local). Precision, Recall and F-measure were computed using the true

Table 5. Criteria for a decision enabling snippet

Person class	Snippet class	Relevant iff
Mobile	Mobility trace	Always (no exception)
Mobile	Destination country trace	There is also a home country trace in the top-5
Mobile	Home country trace	There is also a destination country trace in the top-5
Local	Home country trace	Always (no exception)

mobility status of the people. Kappa was computed for each pair of users sharing the same dataset. A first experiment on automatic person classification is presented as well.

4.6 Automatic MTC Task Evaluation

We performed an automatic person classification test on a set of 25 mobile and 25 local persons. The test consisted in classifying automatically a person as being mobile or local based on geographical criteria, and comparing *Unoporuno* results with the real mobility classes.

5 Results

From results in Table 6 we can observe an $F=0.93$ for the name filtering process, and an $F>0.80$ for all the semantic features. While we can expect to get a very high F when simply controlling for snippet links to a LinkedIn page, more complex features, like biographical sentences, academic degree or organization get a fairly good score. However, further evaluation is needed to measure the impact of the semantic feature analysis on the overall task (see Table 9). Table 7 presents the results of the compared evaluation of statistical classifiers of snippets. The tests were performed on binary trained classifiers (a snippet can be a mobility trace or no trace at all) that selected the top-5 according to the confidence of the prediction. The best score was obtained by the SVM classifier, whose difference with the baseline score is statistically significant ($p < 0.05$). Table 8 presents the

Table 6. Name matching and semantic features evaluation (tp=true positives; fp=false positives; fn=false negatives; P=precision; R=recall; F=F-measure)

Feature	tp	fp	fn	P	R	F
Name matching filter	99	1.00	13	0.99	0.88	0.93
PhD tesis	47	3	5	0.94	1.00	0.97
LinkedIn	50	0	0	1.00	1.00	1.00
Publication	50	0	8	1.00	0.86	0.93
e-mail	46	4	0	0.92	1.00	0.96
Non Latin-American nat.	33	17	0	0.66	1.00	0.80
Latin-American nat.	48	2	1	0.96	0.98	0.97
Person name in URL	47	3	0	0.94	0.90	0.92
CV	48	2	3	0.96	0.94	0.95
Academic degree	47	3	1	0.94	0.98	0.96
Profession	48	2	4	0.96	0.92	0.94
Biographical sentence	49	1	1	0.98	0.98	0.98
Organization acronym	44	6	5	0.88	0.90	0.89
City	40	10	3	0.80	0.93	0.86
Country	44	6	2	0.88	0.96	0.92
Organization	47	3	10	0.94	0.82	0.88
Feature count	-	-	-	-	-	-

Table 7. Top-5 automatic evaluation on the testing set. Classifiers were trained on a binary basis; no geographical data was used for top-5 selection.

Classifier	Top-5 snippets			Persons	
	P@5	R@5	F@5	ODR	mODR
Baseline	0.46	0.08	0.14	0.82	0.72
Dsc. trees	0.32	0.09	0.15	0.76	0.68
Naive Bayes	0.37	0.11	0.17	0.82	0.76
NBtree	0.32	0.11	0.16	0.78	0.72
SVM	0.48	0.13	0.20	0.88	0.84

Table 8. MTC task evaluation on the test set with 7 sociologist users (SVM classifier). Average $F=0.79$ for the first six evaluators (sets A,B,C). Pers. = number of persons in dataset. *Set D corresponds to the full test set.

Data	Pers.	Users	First user			Second user			κ
			P	R	F	P	R	F	
set A	10	E1, E2	0.83	0.56	0.67	0.89	0.89	0.89	0.31
set B	10	E3, E4	0.75	0.75	0.75	0.88	0.78	0.82	0.81
set C	10	E5, E6	0.75	0.75	0.75	0.89	0.89	0.89	0.68
set D*	50	E7	0.80	0.88	0.83	avg. kappa: 0.60			

Table 9. Automatic person classification on the Testing set (50 researchers, SVM classifier)

Id	Ablated feature	P	R	F	F variation
1	ALL FEATURES (no ablation)	0.64	0.78	0.71	-
2	PhD thesis	0.62	0.72	0.67	-0.04
3	LinkedIn	0.64	0.75	0.69	-0.02
4	Publication	0.64	0.72	0.68	-0.03
5	e-mail	0.62	0.78	0.69	-0.02
6	Non Latin-American nat.	0.62	0.75	0.68	-0.03
7	Latin-American nat.	0.64	0.82	0.72	+0.01
8	Person name in URL	0.61	0.83	0.7	-0.01
9	CV	0.59	0.86	0.7	-0.01
10	Academic degree	0.64	0.78	0.71	0
11	Profession	0.64	0.72	0.68	-0.03
12	Biographical sentence	0.61	0.79	0.69	-0.02
13	Organization acronym	0.64	0.72	0.68	-0.03
14	City	0.62	0.75	0.68	-0.03
15	Country	0.64	0.78	0.71	0
16	Organization	0.64	0.78	0.71	0
17	Feature count	0,6	0,78	0,68	-0,03

results of the semi-automatic MTC task carried out by sociologists. An average $F=0.79$ was obtained for the first six evaluators with an average inter-evaluator agreement $\kappa=0.60$ (considered as moderate to substantial). The seventh evaluator annotated the entire test set with an $F=0.83$. From the analysis of the results we observe that a) in approximately 80% of the cases a sociologist

received the right evidence to decide on mobility status; b) the annotator disagreement was higher for the local than for the mobile category (66% agreement for mobile, only 53% for local); c) moderate inter-annotator agreement might be related to low P@5 and R@5: improvements in the snippet classifier could have an impact on this agreement. Finally, an ablation test was made to estimate the impact of each feature on the automatic MTC task. Table 9 shows the result of the automatic classification of 50 persons (25 mobile, 25 local) using all 15 features, which gets an F=0.71, with a mobile recall of 0.78. Further runs were performed, each ablating one feature, in order to estimate the impact of each feature on precision and recall (see Table 9). The features whose absence impacts the overall performance were PhD thesis, Publication, Organization acronym, City, Feature count, Profession and City. In contrast, Organization, Academic degree and Country do not contribute.

6 Conclusion and Further Work

We have shown in this paper how we are using NLP techniques in the sociology of migration field with the *Unoporuno* system. From scientific publication databases, our method produces Web People Search queries refined with bibliographical data, classifies the resulting snippets according to mobility related features and then statistically ranks their relevance. The top-5 snippets are selected for evaluation by a sociologist, and our automatic selection algorithm works in 80% of the cases: using the snippets selected by our system, sociologists can access documents on the Web which allow them to take clear-cut decisions on a person's mobility status with a moderate level of inter-evaluator agreement (avg. kappa=0.60). Furthermore, we presented a first experiment towards automatic annotation of mobility traces. The geographical analysis of locations from snippets classified as mobility traces by an SVM classifier was able to find 78% of the mobile persons of the test set (F=0.71). Further evaluation is necessary to calculate the correlation between sociologists manual annotations and those calculated by *Unoporuno*.

Acknowledgments. Research funded by the CIDESAL Europe-Aid project (MIGR/2008/26). Thanks to Jean-Baptiste Meyer, Martn Koolhaas, Julieta Benegochea, Fernando Esteban, Alejandro Blanco, Jean-Paul Sansonnet and Haydée Lugo for their help.

References

- [1] Auriol, L., Felix, B., Schaaper, M.: Mapping Careers and Mobility of Doctorate Holders: Draft Guidelines, Model Questionnaire and Indicators. OECD Science, Technology and Industry Working Papers (2010/01) (2010)
- [2] Meyer, J.B., Wattiaux, J.P.: Diaspora Knowledge Networks; Vanishing Doubts and Increasing Evidence. International Journal on Multicultural Societies. UNESCO 8(1), 4-24 (2006)

- [3] Artiles, J., Borthwick, A., Gonzalo, J., Sekine, S., Amigó, E.: WePS-3 Evaluation Campaign: Overview of the Web People Search Clustering and Attribute Extraction Task. In: Conference on Multilingual and Multimodal Information Access Evaluation, CLEF (2010)
- [4] Artiles, J., Gonzalo, J., Sekine, S.: The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007). ACL (2007)
- [5] Artiles, J., Gonzalo, J., Sekine, S.: WePS 2 Evaluation Campaign: overview of the Web People Search Clustering Task. In: 18th WWW Conference on 2nd Web People Search Evaluation Workshop, WePS 2009 (2009)
- [6] Sekine, S., Artiles, J.: WePS2 Attribute Extraction Task. In: 18th WWW Conference on 2nd Web People Search Evaluation Workshop, WePS 2009 (2009)
- [7] Artiles, J., Gonzalo, J., Amigó, E.: The impact of query refinement in the web people search task. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACLShort 2009, pp. 361–364. Association for Computational Linguistics, Stroudsburg (2009)
- [8] Liu, J., Birnbaum, L., Pardo, B.: Categorizing blogger’s interests based on short snippets of blog posts. In: Shanahan, J.G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D.A., Kolcz, A., Choi, K.S., Chowdhury, A. (eds.) CIKM, pp. 1525–1526. ACM (2008)
- [9] Barr, C., Jones, R., Regelson, M.: The linguistic structure of English Web-search queries. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, pp. 1021–1030. Association for Computational Linguistics, Stroudsburg (2008)
- [10] Li, X.: Understanding the semantic structure of noun phrase queries. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL 2010, pp. 1337–1345. Association for Computational Linguistics, Stroudsburg (2010)
- [11] Marcos, M.C., Gonzalez-Caro, C.: Comportamiento de los usuarios en la página de resultados de los buscadores. Un estudio basado en eye tracking. *El Profesional de la Información* 19(4) (July-August 2010)
- [12] Mateos, P., Longley, P., Webber, R.: El análisis geodemográfico de apellidos en México. *Papeles de Población* (65), 73–103 (2010)
- [13] Padró, L., Collado, M., Reese, S., Lloberes, M., Castellón, I.: FreeLing 2.1: Five Years of Open-source Language Processing Tools. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation, LREC 2010. European Language Resources Association (ELRA), Valletta (2010)
- [14] Bird, S., Loper, E., Klein, E.: *Natural Language Processing with Python*. O’Reilly Media Inc. (August 2009)
- [15] Steinberger, R., Pouliquen, B., Kabadjov, M.A., Belyaeva, J., der Goot, E.V.: JRC-NAMES: A Freely Available, Highly Multilingual Named Entity Resource. In: Proceedings of the International Conference, RANLP 2011, pp. 104–110 (2011)

Semantic Role Labelling without Deep Syntactic Parsing

Konrad Gołuchowski^{1,2} and Adam Przepiórkowski^{2,1}

¹ University of Warsaw

² Institute of Computer Science, Polish Academy of Sciences

kodie@mimuw.edu.pl,

adamp@ipipan.waw.pl

Abstract. This article proposes a method of Semantic Role Labelling for languages with no reliable deep syntactic parser and with limited corpora annotated with semantic roles. Reasonable results may be achieved with the help of shallow parsing, provided that features used for training such shallow parsers include both lexical semantic information (here: hypernymy) and syntactic information.

Keywords: argument identification, semantic role classification, shallow parsing, chunking.

1 Introduction

Semantic Role Labelling (SRL) is a well-known task within semantic analysis. The idea is to annotate predicate arguments in the sentence with special labels to indicate the semantic relation between the argument and the verb. In the sentence below, three semantic roles (i.e. **Buyer**, **Goods** and **Time**) are indicated.

[**Buyer** Frank] **bought** [**Goods** a new car] [**Time** yesterday].

While much SRL work has been done for English, hardly any is reported for Polish, a language with only prototype-quality parsers and without large or balanced corpora annotated at the level of semantic labels. Moreover, unlike English, Polish is a language with rich inflection and relatively free word order. The experiments reported here were conducted on a small (*83,000-word*) corpus of transcribed phone conversations concerning public transportation in Warsaw, the so-called LUNA corpus [10]. Therefore, annotated situations and semantic roles are limited. This paper proposes a method of semantic labelling in just such setups: without deep syntactic parsing and with a very limited manually annotated corpus.

2 Related Work

In [20], the SRL task is divided into three subtasks: argument identification, semantic role classification and joint annotation. The current paper focuses on the first two: finding phrase boundaries and assigning roles to argument phrases.

Argument identification typically consists of syntactic parsing, followed by binary classification of parse nodes according to whether they represent arguments of predicates [3], perhaps with additional heuristics [21]. Supervised machine learning is also used in the task of semantic role classification, with much work devoted to feature selection [3,2,19,14].

Few papers propose methods not involving deep parsing. [6] uses SVM to assign to syntactic chunks IOB tags derived from semantic roles. [18] uses shallow parsing for Chinese SRL. [15] compares systems that use deep and shallow parsing and shows implications of the lack of complete trees for argument identification.

3 SRL without Complete Syntactic Parsing

Most approaches mentioned above make significant use of the syntactic parse tree. Many features for classification and argument identification heuristics are based on spans of tree nodes and relations between them. Such features are designed to reflect some relation between the syntactic realization of arguments and their semantics. The lack of a syntactic parse also makes argument identification task much harder: since semantic arguments are normally realised as syntactic constituents, obtaining the tree of a sentence is paramount to constraining the set of candidates for arguments. Finally, information contained in the syntactic tree helps to decide which of possibly several predicates in the sentence governs a given argument.

To alleviate the problem of no complete syntactic parse information, shallow parsing is applied here to extract basic syntactic (nominal, prepositional, etc.) groups (roughly: chunks [1]), using a shallow grammar (a cascade of regular grammars) manually developed within a different project [54]. Initial experiments indicate that a certain level of correlation exists between such groups and predicate arguments. Treating these groups as arguments yields acceptable recall but very low precision:

Arguments match criteria	Precision	Recall	F-measure
Exact match	0.15	0.44	0.22
Overlap	0.29	0.86	0.43

Given these unsatisfactory results, an *additional* approach to shallow parsing was implemented, based on IOB tagging [16] and taking advantage of both syntactic and semantic features. The tagger trained here implements the linear-chain Conditional Random Fields model.

The key problem was the selection of features relevant for identifying argument boundaries. The following features have been used to train the model: 1) word shape (e.g. *Ul* for “Desk”, where *U* stands for a sequence of upper-case letters and *l* stands for a sequence of lower-case letters), 2) the most general hypernym of the word (based on Polish WordNet, [13,9]) if available and base form of the word otherwise, 3) the word’s part of speech, 4) the word’s case (if relevant), 5) the syntactic group (if the word is contained inside one) identified by the shallow grammar, 6) all above features for the two immediately adjacent words.

4 Semantic Role Classifier

After argument identification, the next step is to assign a semantic role to each argument. The MaxEnt-based classifier implemented in the SciKit package [12] was trained for this purpose.

As in the argument identification task, many commonly used features in this task could not be used due to the lack of a parse tree. In particular, it is not known which of possibly many predicates actually governs a given argument. Instead, the closest potential predicates to the left and to the right of the argument are identified, and the following features are used to decide what semantic role the predicate assigns to the argument: 1) the base form of each predicate, 2) their parts of speech (PoS), 3) whether predicates are negated, 4) the type of the syntactic group (if overlapping with argument boundaries), 5) the case of the noun in the argument (if applicable), 6) the left-most preposition in the argument, 7) PoS of the first and last words in the argument, 8) the most general hypernyms of the words from argument available in the WordNet, and the base form of the word otherwise, 9) the words' prefixes and suffixes of length three.

Originally, the LUNA corpus was annotated with 64 FrameNet-like roles [8]. However, more than half of these roles occurred less than 15 times in the corpus. Therefore, FrameNet-like roles were semi-automatically projected to 19 more general VerbNet-based thematic roles [17] utilizing Semlink resources [11].

5 Evaluation

When determining whether a given argument has been identified correctly, one may require the complete identity of spans or loosen this requirement to mere overlap. Both approaches have been used in the past. A compromise but still relatively strict approach is proposed here: the potential argument is judged as correctly identified if it differs from the gold standard argument at most with respect to initial or final potential modifiers (i.e., particles, adjectives, etc.), as in the English NPs *books about Bali* vs. *even these books about Bali* (containing the focus particle *even* and the demonstrative *these*).

The usual 10-fold cross-validation was performed. For the classification of semantic roles, two sets of results are given: for role classification as a separate task assuming prior gold-standard argument identification (Table 1) and as a joint task with argument identification (Tables 2 and 3). Note that the results in Table 1 are optimistic, as they assume not only prior identification of arguments, but also – in the last row – the correct identification of the predicate governing the argument. This table also contains a comparison of semantic role classification using FrameNet-like roles and thematic VerbNet-like roles.

In order to measure the importance of particular syntactic and semantic features introduced in Section 4, experiments were repeated with different feature sets. The most significant results are presented in Table 4, which expands the 2nd row of Table 1.

Table 1. Results of semantic role labelling on gold standard arguments

Task	Accuracy
Semantic role labelling (FrameNet roles)	0.65
Semantic role labelling (VerbNet roles)	0.74
Semantic role labelling (VerbNet roles) + correct predicate	0.76

Table 2. Results of argument identification task and semantic role classification task with proposed compromise solution for arguments' agreement

Task	Precision	Recall	F-measure
Argument identification	0.71	0.68	0.70
Arg. identification + semantic role classification	0.61	0.57	0.59

Table 3. Results of argument identification task and semantic role classification task when identity of argument spans is required

Task	Precision	Recall	F-measure
Argument identification	0.69	0.64	0.67
Arg. identification + semantic role classification	0.58	0.54	0.56

Table 4. Results of semantic role classification on gold standard arguments with different set of features

Features	Accuracy
Predicate features(1,2,3) + Case(5)	0.55
Predicate features(1,2,3) + Preposition(6)	0.57
Predicate features(1,2,3) + Syntactic group(4) + Case(5)	0.60
Predicate features(1,2,3) + Case(5) + Preposition(6)	0.62
Predicate features(1,2,3) + All syntactic features(4,5,6,7)	0.66
Predicate features(1,2,3) + Hypernyms(8)	0.63
Predicate features(1,2,3) + Case(5) + Preposition(6) + Hypernyms(8)	0.71
Syntactic group(4) + Case(5) + Preposition(6) + Hypernyms(8)	0.67
Predicate's PoS(2) + Syntactic group(4) + Case(5) + Preposition(6) + Hypernyms(8)	0.67
Predicate's lemma(1) + Syntactic group(4) + Case(5) + Preposition(6) + Hypernyms(8)	0.73
All features (1-9)	0.74

Due to the fact that arguments are usually adjective, noun or prepositional phrases, syntactic features are crucial in the task of argument identification as they are of great help in extracting such phrases. Moreover, predicates very often impose certain restrictions on syntactic features of semantic roles. In fact, case

(5) and preposition (6) presents high correlation with semantic role occurrences and are very useful in semantic role classification. Only the base forms of predicates seem important in case of predicate features. Also, one can observe that the use of hypernyms (from the Polish WordNet) greatly improves the results.

6 Summary and Future Work

This article presents initial experiments with semantic role labelling for Polish. It proposes a method that does not require a syntactic parse tree to identify arguments and instead relies on the output of the shallow parser both for argument identification and for semantic role classification. Evaluation includes the impact of different features on the accuracy of semantic role classification.

Because the IOB tagger produces arguments that do not overlap, there is no need for joint annotation, used for example in [20]. However, future work should examine the possibility of increasing argument identification recall by introducing overlapping arguments. The step of joint annotation also gives the opportunity to take advantage of probabilities of various semantic roles for each argument.

During the experiments it turned out that the corpus employed here is rather noisy, apart from being small and unbalanced. In order to build a robust SRL system, a bigger and cleaner corpus is needed. One possibility to build such a corpus is to exploit the existence of parallel corpora and SRL tools for English (see [7]).

References

1. Abney, S.: Parsing by chunks. In: Berwick, R., Abney, S., Tenny, C. (eds.) *Principle-Based Parsing*, pp. 257–278. Kluwer (1991)
2. Fleischman, M., Kwon, N., Hovy, E.: Maximum entropy models for framenet classification. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2003)
3. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. *Computational Linguistics* 28(3), 245–288 (2002)
4. Głowińska, K.: Anotacja składniowa NKJP. In: Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B. (eds.) *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw (2012)
5. Głowińska, K., Przepiórkowski, A.: The design of syntactic annotation levels in the National Corpus of Polish. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC 2010*. ELRA, Valletta (2010)
6. Hacioglu, K., Pradhan, S., Ward, W., Martin, J.H., Jurafsky, D.: Semantic Role Labeling by Tagging Syntactic Chunks. In: *Proceedings of CoNLL 2004*, pp. 110–113 (2004)
7. Johansson, R., Nugues, P.: A FrameNet-based semantic role labeler for Swedish. In: *Proceedings of the COLING/ACL on Main Conference Poster Sessions, COLING-ACL 2006*, pp. 436–443. Association for Computational Linguistics, Stroudsburg (2006)

8. Johnson, C.R., Fillmore, C.J., Petruck, M.R., Baker, C.F., Ellsworth, M.J., Ruppenhofer, J., Wood, E.J.: *FrameNet: Theory and Practice* (2002)
9. Maziarz, M., Piasecki, M., Szpakowicz, S.: Approaching plWordNet 2.0. In: *Proceedings of the 6th Global Wordnet Conference*, Matsue, Japan (2012)
10. Mykowiecka, A., Marasek, K., Marciniak, M., Rabięga-Wisniewska, J., Gubrynowicz, R.: Annotated Corpus of Polish Spoken Dialogues. In: Vetulani, Z., Uszkoreit, H. (eds.) *LTC 2007. LNCS*, vol. 5603, pp. 50–62. Springer, Heidelberg (2009)
11. Palmer, M.: SemLink—linking PropBank, VerbNet, FrameNet and WordNet. In: *Proceedings of the Generative Lexicon Conference*, Pisa, Italy (2009)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 2825–2830 (2011)
13. Piasecki, M., Szpakowicz, S., Broda, B.: A Wordnet from the Ground Up. *Oficyna Wydawnicza Politechniki Wrocławskiej*, Wrocław (2009)
14. Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J.H., Jurafsky, D.: Support vector learning for semantic argument classification. *Machine Learning* 60(1-3), 11–39 (2005)
15. Punyakanok, V., Roth, D., Yih, W.T.: The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics* 34(2), 257–287 (2008)
16. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: *Proceedings of the Third Workshop on Very Large Corpora*, pp. 82–94. ACL, Cambridge (1995)
17. Schuler, K.K.: *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania (2006)
18. Sun, W., Sui, Z., Wang, M., Wang, X.: Chinese semantic role labeling with shallow parsing. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009*, vol. 3, pp. 1475–1483. Association for Computational Linguistics, Stroudsburg (2009)
19. Surdeanu, M., Harabagiu, S., Williams, J., Aarseth, P.: Using predicate-argument structures for information extraction. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, ACL 2003*, vol. 1, pp. 8–15. Association for Computational Linguistics, Stroudsburg (2003)
20. Toutanova, K., Haghighi, A., Manning, C.D.: A global joint model for semantic role labeling. *Computational Linguistics* 34(2), 161–191 (2008)
21. Xue, N.: Calibrating features for semantic role labeling. In: *Proceedings of EMNLP 2004*, pp. 88–94 (2004)

Temporal Information Extraction with Cross-Language Projected Data

Przemysław Jarzębowski¹ and Adam Przepiórkowski^{2,1}

¹ University of Warsaw

² Institute of Computer Science, Polish Academy of Sciences
p.jarzebowski@students.mimuw.edu.pl,
adamp@ipipan.waw.pl

Abstract. This paper presents a method used for extracting temporal information from raw texts in Polish. The extracted information consists of the text fragments which describe events, the time expressions and the temporal relations between them. Together with temporal reasoning, it can be used in applications such as question answering or for text summarization and information extraction. First, a bilingual corpus was used to project temporal annotations from English to Polish. This data was further enhanced by manual correction and then used for inducing classifiers based on Conditional Random Fields (CRF) and a Support Vector Machine (SVM). For the evaluation of this task we propose a cross-language method that compares the system's results with results for different languages. It shows that the temporal relations classifier presented here outperforms the state of the art systems for English when using the macro-average F_1 -measure, which is well suited for this multiclass classification task.

Keywords: temporal information, temporal relation, event extraction, word alignment.

1 Introduction

One of the key elements of deep text understanding is the ability to process temporal information. Those parts of natural language texts which describe sequences of events often mention times of occurrence of these events. Being able to establish such a temporal relation between events and their occurrence times just by analysing a sentence would much enhance some NLP applications. Temporal reasoning, for example, is an essential part of many question answering systems. Information about events and their time of occurrence automatically extracted from sources such as news articles or Wikipedia would make it possible to answer a broad range of time-related questions. Furthermore, it would make it possible to infer relations between events. In text summarization, knowledge about the events mentioned might be a good indicator of the text's most significant or informative parts.

Depending on the target application of the extracted temporal information, the definition of an event can differ, and temporal relations can take a different set of values. TimeML [13] together with annotation guidelines created for TimeBank corpora [14] present a formalization of the temporal information extraction task. They specify which fragments of the text should be identified as events, time expressions and also defines types of temporal relations. This commonly used standard ([8],[11],[15]) is followed here.

The process of extracting temporal information can be split into three tasks: identification of time expressions, identification of events and classification of temporal relations between time expressions and events. Training supervised machine learning classifiers to solve the last two of them has proven to give the best results [17]. However, this method requires data with temporal annotation, which for Polish was not available. Manually creating annotation is an expensive and long process, so instead, a bilingual corpus and word alignment were used to project annotations from English to Polish. An example of a sentence with its temporal information projected from English to Polish is presented in Table 1.

The annotation for the English part of the bilingual corpus was created automatically by TIPSem [8] – a temporal information system for English. Next, the extracted events, time expressions and temporal relations between events and time expressions were projected to the Polish part of the corpus. The annotation obtained this way is noisy not only because of word alignment errors, but also because of misclassifications by TIPSem. Projection constraints were applied to limit both types of errors. The scarcity of errors in the projected events makes them acceptable to use as reference data for the further process, but the classified temporal relations contain relatively more errors. A part of them was manually corrected and the rest was used only to boost the classifier. Note that this work is only concerned with temporal relations between events and time expressions, and the annotation of such relations is much less time consuming than complete temporal annotation involving event annotation. The projection algorithm and the correction process are described in detail in Sec. 3.

Classifying the type of temporal relation between different events is a difficult task. When annotating the TimeBank corpus, the inter-annotator agreement on the type of temporal relation was $F_1 = 0.55$ [11]. The authors of the article argued that this low score was due to the large number of event pairs available for comparison, so it was difficult for annotators to spot all of the existing temporal relations. This low score has a great impact on the quality of the data used to build classifiers. In order to avoid this problem, the work presented here considers only the classification of temporal relations between events and temporal expressions, where the data is much more reliable, and does not consider temporal relations between events. Also, these temporal relations are much more significant for some of the applications mentioned above because one could use them to accurately put the events on a timeline.

The dataset thus created was used to induce two classifiers: an event classifier and a classifier of temporal relations. The first one uses Conditional Random

Fields (CRF [7]), which is also used by TIPSem; the latter is based on Support Vector Machine (SVM [3]). Details of the training of the classifiers and the text features used are described in Sec. 4. Unlike English, Polish does not enforce strict word order in sentences and is highly inflectional. This has a great impact on the features chosen for the classification. For extracting time expressions, a set of extraction rules was created and a rule-based shallow parsing system Spejd [4] was used. The defined extraction rules use both the lemma of a word and its morphosyntactic properties.

Section 5 presents the results of the evaluation of the event classifier and the temporal relation classifier. By projecting temporal relations across languages, a comparison is made between the results for the classification of temporal relations obtained here and the results of Evita [15] and TIPSem.

2 Related Work

Application of the TimeML standard makes it possible to compare the results achieved here with those reported for state of the art systems, specifically the Evita system and the system that had the best score in the TempEval2 competition [17] in the events identification task – TIPSem. Evita integrates a rule-based approach with machine learning for event recognition and classification of temporal relations. TIPSem is based on machine learning, and the set of features it uses for classification is enriched with semantic roles. Just as in case of TIPSem, the current work follows the machine learning approach.

In the solution presented here, a word-aligned bilingual corpus was used to create a resource with temporal annotation in a new language, similarly to [16]. There, temporal annotation was projected from English to German to build classifiers for events, time expressions and temporal relations. On the other hand, in the work presented here, the projected data is used for inducing an event classifier, for boosting the classifier of temporal relations, but also to perform a cross-language comparison of systems. This comparison was based on the manual annotation of a small set of the temporal relations in the word-aligned bilingual corpus.

In comparison with TimeEval2 and its task of temporal relation classification, the work presented here focuses not only on assigning a temporal relation type but also on making a decision whether the temporal relation exists or not, as in [10]. In our work, for the purpose of evaluating the classified temporal relations, a macro-average F_1 measure is reported. Macro-average F_1 is the average of the F_1 scores computed for each of the types of temporal relation. Some of the types of temporal relations are much more frequent than others, and this measure ensures that the ability of the resulting classifier to classify all of them with high performance is included in the final score. To the best of our knowledge, this problem of the minority relation types was not addressed in previous work.

3 Creating Temporal Data

The annotation of events in the Polish part of the corpus was obtained by projecting annotations from the English part. For this purpose, word alignment between the two parts of a parallel corpus was performed. An example of the projection of temporal information using the word alignment from Fig. 1 is presented in Table 1.

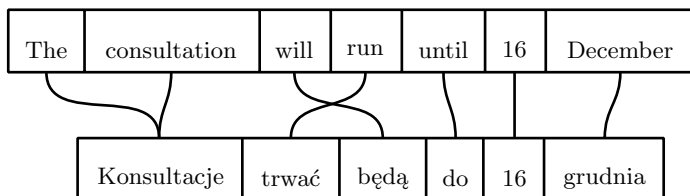


Fig. 1. Example of word alignment

Table 1. Example of temporal information projection using word alignment shown in Fig. 1

Sentence	<i>The</i> [<i>consultation</i>] _{event₁} <i>will</i> [<i>run</i>] _{event₂} <i>until</i> [<i>16 December</i>] _{time_{x1}} .
annotation:	[<i>Konsultacje</i>] _{event₁} [<i>trwać</i>] _{event₂} <i>będą</i> <i>do</i> [<i>16 grudnia</i>] _{time_{x1}} .
Temporal relations:	[<i>event₁</i>] ENDED-BY [<i>time_{x1}</i>] [<i>event₂</i>] ENDED-BY [<i>time_{x1}</i>]

3.1 Developing Word Alignment

Given one sentence written in two languages – a source sentence and its translation – word alignment, which looks at pairs of words across languages, finds those which have the corresponding meaning. It is not always a one-to-one relationship, and often for one English word multiple Polish words are found, and the other way around. The reasons for this are grammar differences between Polish and English (e.g. no determiners or phrasal verbs in Polish), and lexical differences (e.g. idioms).

Community Research and Development Information Service (CORDIS¹) parallel corpus was used, and the word alignment was created for 146,334 of its sentences with a statistical machine translation tool – Moses [6]. Moses first builds a Hidden Markov Model for the entire corpus, and then for each sentence chooses the alignment with the highest probability, computing it with the built translation model. The final alignment is the result of merging two separate alignments: the Polish-English alignment and the English-Polish alignment. Both of them are of the type one-to-many. Merging them gives a many-to-many alignment reflecting the true relationship between words in those languages.

¹ <http://cordis.europa.eu>

Also, before the alignment was computed, the Polish part of the corpus was preprocessed, and all words were substituted for their lemmas using the tools Morfeusz [18] and Pantera [1]. The positive impact of this preprocessing step on the alignment accuracy was presented in [19]. Polish is a highly inflectional language, and the number of unique word types is much higher in the Polish part of the parallel corpus than it is in the English part. Because creating the word alignment is based on a simple string comparison, before the lemmatization all of the inflections of one Polish lemma were treated as different words. Lemmatization considerably increased the frequency of Polish-English word pairs, which helped in increasing the accuracy of the computed word alignment.

3.2 Projecting Annotation

The developed word alignment was next used to project temporal information from English to Polish. **EVENT** and **TIMEX** tags, which cover the extent of events and the extent of time expressions, were copied alongside the word alignment. If the alignment was of the type one-to-many then the tag was multiplied. With the tags also their identifiers assigned by TIPSem were projected and, as a result, the temporal relations found by TIPSem became valid for the Polish part of the corpora. An example of a projection is shown in Table 1.

In order to limit the number of incorrect projections some constraints were applied. Some of them were suggested in [16]. Tag projections were required to be a contiguous sequence of words, and they were not allowed to clash, for example if two different events were projected to the same word. Unlike [16], the constraint on the **TIMEX** tags that they contain only content-bearing words (tokens which are not prepositions or punctuation) was not applied here. Prepositions, for example, are a valid part of a time expression, especially in the case of time expressions describing a duration, e.g. *from January to March 2011*. Sentences in which any of the constraints was not fulfilled were discarded from the dataset. Also, all the sentences which, after the projection, did not have any **EVENT** tag were omitted.

3.3 Annotating Temporal Relations

The temporal relations between events and time expressions were assigned one of the types: *before*, *ended by*, *after*, *begun by*, *is included*. *Is included* means that the event has happened in the time period defined by the given time expression. The TimeML annotation guidelines propose a total of 14 different temporal relations. However, some of them were symmetrical to the ones chosen here, and the other ones were not found useful for the aims of the presented work, i.e. finding temporal relations between events and time expressions. Also, TIPSem and Evita use the same types to describe temporal relations, but they also use the *simultaneous* type which is a special case of *is included*. During the annotation process, it was enforced that all the pairs of events and time expressions in a sentence were assigned one of the defined types, or *none* if there is no temporal relation. This addressed the problem of annotators accidentally omitting some of the temporal relations in the sentence. Those negative examples of temporal

relations were also used when inducing the classifier, so that it is able to discriminate between the existence and non-existence of a temporal relation.

The types of temporal relations are unevenly distributed in text. The most frequent case is that there is no temporal relation between the given event and time expression. The relations *begun by*, *ended by* are rare (e.g. the TimeBank corpus for about 6.5K temporal relations has less than 1% relations *begun by*). In order to build a more balanced dataset, the preliminary projected temporal relations were used as a cue of their actual types. 187 sentences with 606 TIMEX–EVENT pairs were selected for manual annotation.

The events and time expressions used during the manual annotation are a result of the projection of temporal information found by TIPSem for English text, and they can contain some errors. A sentence with its TIMEX–EVENT pairs was discarded from the temporal relations corpora if the annotator discovered that an event which is in a temporal relation with the given time expression was not automatically detected. This way the annotators’ work was limited to assigning a temporal relation rather than correcting the projected annotation and finding the actual events in the sentence. As a result, 240 temporal relation instances other than **none** were obtained, which is 4 times less examples than there were available for training in the TempEval2 contest for this task. That data was used both for training the temporal relations classifier and for evaluating the quality of data obtained with TIPSem.

4 Classification

4.1 Event Recognition

In the literature two different approaches to the task of event recognition were proposed: one involved building a vocabulary of words describing events (e.g. [2]), and the other approach used a machine learning classifier. One of the problems with the first approach is the property of language that a word which in one context describes an event, does not necessarily do so in another context. An example using the word *discover* is shown in the two sentences below:

- *The discovery of penicillin in 1928 by A. Fleming was a great breakthrough in medicine.*
- *The discovery of a cure for cancer would be a great breakthrough in medicine.*

For this reason a machine learning approach was applied which in comparison with using a set of defined rules can be much more resilient in such cases. The remaining part of this section introduces the method which is based on Conditional Random Fields.

Conditional Random Fields is a machine learning approach for sequence classification. Its main feature is that the classification process can use information about the class already assigned to the previous element in a sequence. In this application of CRFs the sequences are words arranged by their order of

appearance in a sentence. Usually, two different events do not occur in a sentence as consecutive words, and if they do, they are a sequence of words with a specific relationship between them [9], as in *begin meeting*. This knowledge can be incorporated by CRFs classifier.

Text Features. NLP for Polish is at a much less developed stage than it is for English, for example there is no robust syntactic parser available. For this reason, to represent the context features of words, the output of Spejd [4] – a rule-based shallow parsing tool – was used. Extraction rules were adopted to identify syntactic categories such as noun phrases and prepositional phrases. Also, morphosyntactic features such as the case of a word were used, because they carry some information about the word’s semantic role in a sentence. For example the accusative case of a noun can mean that it is a patient of a verb. The features of words used for event recognition are:

- Lemma
- Polish WordNet [12] hypernyms of a lemma
- Grammatical features – part of speech (POS), case, gender, voice, tense, aspect, number
- Spejd features – syntactic category of a word, e.g. noun phrase, adjective phrase, adverb phrase, prepositional phrase
- Temporal expression proximity – often events are in close proximity to a time expression in a sentence. Those features contain information about whether the word is in the same dependent clause as a time expression, whether it occurs before or after one, and information about its distance from the time expression measured in number of words.

4.2 Temporal Relations

The goal of the temporal classifier is to decide whether there is a temporal relation between the given time expression and event, and if there is one, then to classify its type. For this purpose, features of events and time expressions are used, as well as information about their relative position in a sentence. When developing the classifier the average of the individual F_1 values for all of the class types was maximised in a cross-validation process, so the resulting classifier can detect and classify a temporal relationship equally well regardless of its type.

SVM, which gives the best results for many machine learning tasks, was chosen here as a classification method. Each of the TIMEX–EVENT pairs was classified separately, and for that classification we did not find a strong use case for CRF classifier which can incorporate classification results of other data examples. Sampling was used when choosing the training data for each of the folds in order to guarantee that each of the data classes were of similar sizes.

Also a heuristic was implemented, which is applied if, for a given time expression, the classifier does not discover in the sentence any event with which that time expression is in a temporal relation. The heuristic is motivated by the assumption that at least one event in the sentence must be in a temporal

Table 2. Event recognizer results against baselines

Classifier	Precision	Recall	F_1
Verbs only	51.3	70.3	59.3
Verbs + nouns selected with WordNet	42.1	82.7	55.8
Event recognizer	77.4	66.5	71.5

Table 3. Event recognizer results by POS

POS	Precision	Recall	F_1
Verb	79.9	83.1	81.5
Noun	62.4	32.6	42.8
Adjective	72.7	2.3	4.4
Other	75.0	7.1	13.0

relation with each time expression, i.e. dates in a sentence always describe the time of some event. Among the **TIMEX**–**EVENT** pairs with that time expression a pair which has the highest probability of some not *none* temporal relation type is found, and that temporal relation is assigned to that pair.

The following features were used for training the classifier of temporal relations:

- Features of time expressions – information about whether the time expression describes a specific date or a period of time, and prepositions preceding the time expression. Prepositions such as *after*, *while*, *until* often indicate the type of a temporal relation as is shown by experiments in [5].
- Features of events – morphological features of a head word describing the event, its tense and information about the syntactic category to which belongs.
- Features describing the relative position between events and time expressions – information about whether the word is in the same dependent clause as a time expression, whether it occurs before or after one, information about its distance from the time expression measured in number of words, and information about the syntactic categories on a path between the event and the time expression.

5 Evaluation

5.1 Event Recognition

The evaluation of the event recognizer was conducted with 5-fold cross-validation on the projected data. Because this data was automatically obtained with TIPSem and word alignment, it is not free from misclassifications. These results are reported below in order to give an indication of how well the classifier is able to replicate TIPSem results. Table 2 compares the classifier’s result with simple baselines which classify words as events using their POS and WordNet classes. Table 3 presents results broken down according to the POS of words denoting events.

Very low recall for adjectives can be explained by the fact that the dataset contains far less examples of events described with an adjective than with a noun or a verb. TIPSem performs worse in annotating those as well, which has an impact on the quality of the training and testing data.

5.2 Temporal Relations

The evaluation of temporal relations was conducted with 5-fold cross-validation using manually annotated temporal relations (536 instances). Also, for the training of the classifier, a small sample of the automatically projected temporal relations was used in order to boost the classifier.

When reporting the performance of the classifier, the accuracy of the classifier’s decision on whether the temporal relation exists or not is also considered. Although this was not assessed in the TempEval2 contest, we think that discriminating between these two situations is as important as deciding on the type of temporal relation. As well as the accuracy measure, the G_{mean} average and the average of the F_1 scores is reported for all the classes. G_{mean} is a geometrical mean of all the classes’ recalls, and is frequently used to assess the quality of results when dealing with data unevenly distributed between classes. Maximising the average of the individual F_1 values guarantees that precision for all of the temporal relation types, as well as recall, will be included in the final score.

A baseline classifier following a simple algorithm was developed to compare the results obtained. The baseline for each time expression in a sentence chooses the event which is closest and assigns the temporal relation type based on the preposition before the time expression. If there is no preposition or the time expression has a duration type, then the *is included* relation is assigned. All other TIMEX–EVENT pairs with that time expression are assigned *none*. The results of the comparison are presented in Table 4.

Table 4. Results for the temporal relation classification

Classifier	Accuracy	G_{mean}	$F_1 avg$
Baseline	74.7	43.8	53.4
Temporal relation wo heuristics	78.5	59.3	60.1
Temporal relation classifier	79.0	58.3	62.8

The manually annotated temporal relations were also used to compare the performance of the classifier presented here with Evita and TIPSem. The events and time expressions used for annotation come from the projection of the temporal information found by TIPSem, so comparing its results with ours is straightforward. To compare the temporal relations found by Evita a mapping of its events and time expressions onto those found by TIPSem was applied. Some of the events were not recognised by Evita and vice versa, and as a result 276 events which matched the manually annotated temporal relations were found.

By comparing the results across languages an assumption is made that the Polish translations in the parallel corpus does not change the type of temporal relation. This unfortunately is not always true, and some translations not following this rule were found. The comparison is presented in Table 5. The low G_{mean} and $F_1 avg$ score of TIPSem is due to its low performance for minority types of temporal relations, especially *after* and *before*.

Table 5. Comparison of the results for the classification of temporal relations across languages

Classifier	Accuracy	G_{mean}	$F_1 avg$
TIPSem	73.7	0.0	24.4
Evita	66.7	35.2	44.6
Polish classifier	79.0	58.3	62.8

6 Conclusions and Future Work

This paper presents an approach to temporal information extraction from texts in languages which do not have dedicated corpora with temporal annotation. It uses the word alignment technique from the field of machine translation to create the required resources and then applies machine learning methods to train the event recognizer and the temporal relations classifier. The approach from [16] is extended here, and the manually annotated temporal relations are also used to directly compare performance of the presented system with the state of the art systems for English. This work shows that the effort to create resources in different languages for the task of temporal relations classification can benefit all of them. The manually annotated data in the Polish part of a parallel corpus can be projected to English and used as data for comparison of the systems.

In the presented work it is also proposed to maximise the macro-average F_1 measure when training a temporal relations classifier. This ensures that even the less frequent types of relation are classified with high performance.

The results of the evaluation show that the classification of temporal relations between EVENT – TIMEX pairs for Polish can be performed with relatively high accuracy just using prepositions. The applied machine learning approach which uses shallow parsing features of text improves that baseline, and significantly outperforms the temporal relations obtained by projecting data annotated with TIPSem and Evita. Those results suggest that either the task itself is easier for Polish language because of clearer relations between the preposition and the type of the temporal relation, or that the state of the art systems did not perform well in the classification of the minority types of temporal relations.

Future work should focus on enriching the corpus with manually annotated temporal relations, so that the less frequent types of temporal relations are represented by more examples. This could significantly increase the performance of their classification, as very few relations of those types were found automatically by the English systems.

References

1. Acedański, S.: A Morphosyntactic Brill Tagger for Inflectional Languages. In: Loftson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) *IceTAL 2010*. LNCS, vol. 6233, pp. 3–14. Springer, Heidelberg (2010)
2. Arnulphy, B., Tannier, X., Vilnat, A.: Automatically Generated Noun Lexicons for Event Extraction. In: Gelbukh, A. (ed.) *CICLing 2012, Part II*. LNCS, vol. 7182, pp. 219–231. Springer, Heidelberg (2012)
3. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152. ACM Press (1992)
4. Buczyński, A., Przepiórkowski, A.: Spejd: A Shallow Processing and Morphological Disambiguation Tool. In: Vetulani, Z., Uszkoreit, H. (eds.) *LTC 2007*. LNCS, vol. 5603, pp. 131–141. Springer, Heidelberg (2009)
5. Derczynski, L., Gaizauskas, R.: Using signals to improve automatic classification of temporal relations. In: *Proceedings of the ESSLLI StuS (2010)*
6. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 2007*, pp. 177–180. Association for Computational Linguistics, Stroudsburg (2007)
7. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning, ICML 2001*, pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco (2001)
8. Llorens, H., Saquete, E., Navarro, B.: TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 284–291. Association for Computational Linguistics, Uppsala (2010)
9. Llorens, H., Saquete, E., Navarro-Colorado, B.: TimeML events recognition and classification: learning CRF models with semantic roles. In: *Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010*, pp. 725–733. Association for Computational Linguistics, Stroudsburg (2010)
10. Llorens, H., Saquete, E., Navarro-Colorado, B.: Automatic system for identifying and categorizing temporal relations in natural language. *International Journal of Intelligent Systems* 27(7), 680–703 (2012)
11. Mani, I., Verhagen, M., Wellner, B., Lee, C.M., Pustejovsky, J.: Machine learning of temporal relations. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL-44*, pp. 753–760. Association for Computational Linguistics, Stroudsburg (2006)
12. Maziarz, M., Piasecki, M., Szpakowicz, S.: Approaching plWordNet 2.0. In: *Proceedings of the 6th Global Wordnet Conference, Matsue, Japan (2012)*
13. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust specification of event and temporal expressions in text. In: *Fifth International Workshop on Computational Semantics, IWCS-5 (2003)*
14. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The TimeBank corpus. In: *Proceedings of Corpus Linguistics 2003*, pp. 647–656 (2003)

15. Sauri, R., Knippen, R., Verhagen, M., Pustejovsky, J.: Evita: A Robust Event Recognizer for QA Systems. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 700–707. Association for Computational Linguistics, Stroudsburg (2005)
16. Spreyer, K., Frank, A.: Projection-based acquisition of a temporal labeller. In: Proceedings of IJCNLP 2008, Hyderabad, India, pp. 489–496 (2008)
17. Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jang, S.B., Littman, J., Rumshisky, A., Phillips, J., Pustejovsky, J.: Automating temporal annotation with TARSQI. In: Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions, ACLdemo 2005, pp. 81–84. Association for Computational Linguistics, Stroudsburg (2005)
18. Woliński, M.: Morfeusz – a Practical Tool for the Morphological Analysis of Polish. In: Kłopotek, M., Wierzchon, S., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. Advances in Soft Computing, vol. 35, pp. 511–520. Springer, Heidelberg (2006), http://dx.doi.org/10.1007/3-540-33521-8_55
19. Wróblewska, A.: Polish-English Word Alignment: Preliminary Study. In: Ryżko, D., Rybiński, H., Gawrysiak, P., Kryszkiewicz, M. (eds.) Emerging Intelligent Technologies in Industry. SCI, vol. 369, pp. 123–132. Springer, Heidelberg (2011)

Word Sense Disambiguation Based on Example Sentences in Dictionary and Automatically Acquired from Parallel Corpus

Pulkit Kathuria and Kiyooki Shirai

Japan Advanced Institute of Science and Technology

{pulkit,kshirai}@jaist.ac.jp

<http://www.jaist.ac.jp/nlp/lab/index.html>

Abstract. This paper presents a precision oriented example based approach for word sense disambiguation (WSD) for a reading assistant system for Japanese learners. Our WSD classifier chooses a sense associated with the most similar sentence in a dictionary only if the similarity is high enough, otherwise chooses no sense. We propose sentence similarity measures by exploiting collocations and syntactic dependency relations for a target word. The example based classifier is combined with a Robinson classifier to compensate recall. We further improve WSD performance by automatically acquiring bilingual sentences from a parallel corpus. According to the results of our experiments, the accuracy of automatically extracted sentences was 85%, while the proposed WSD method achieves 65% accuracy which is 7% higher than the baseline.

Keywords: Word Sense Disambiguation, Machine Readable Dictionary, Example Based Method, Examples Expansion, Parallel Corpus, Japanese.

1 Introduction

Japanese learners often look up words in dictionaries/internet when they read Japanese documents. One word has several possible translations, although a word has only one meaning when it appears in the document. It is rather hard for non-native readers of Japanese to read definition sentences of all meanings. It would be useful to build a system which can not only show the target word's definition sentence in English and its example usage but also select the correct meaning. Currently, ASUNARO¹ is the only reading assistant system for Japanese learners with Word Sense Disambiguation (WSD hereafter) module. However, definition sentences of EDR dictionary² produced by ASUNARO are sometimes unnatural and no example sentence is shown for each sense. In our reading assistant system, we use EDICT³, the Japanese-English bilingual dictionary that includes definition sentences in English as well as example sentences

¹ <http://hinoki.ryu.titech.ac.jp/asunaro/main.php?lang=en>

² <http://www2.nict.go.jp/out-promotion/techtransfer/EDR/index.html>

³ <http://www.csse.monash.edu.au/~jwb/edict.html>

in Japanese and English. We believe that example sentences are indispensable for Japanese learners to understand meanings of words.

WSD in our reading assistant system is a task of translation selection in machine translation. It has been shown in previous researches that lexical translation selection by using WSD helps to improve performance and quality in Machine Translation (MT). Carpuat et al. and Chan et al. integrated WSD in MT and showed significant improvements in terms of general MT quality, on a Chinese-English MT framework [1,2].

In this paper, example-based WSD is considered, since it would be suitable for our reading assistant system showing examples to users. It should handle all words, including low frequency words, in a document. Therefore our WSD method does not rely on a sense tagged corpus that requires much human labor to construct, although many of current WSD methods use supervised machine learning [3,4]. Proposed example based classifier uses EDICT as example database and is designed to choose a sense only in reliable cases, that there is a similar sentence in example database. Such a system would achieve high precision but low recall. To compensate recall, our example based method is combined with a more robust WSD method. Furthermore, we automatically extract bilingual example pairs from a parallel corpus. It enables us to improve the performance of WSD classifiers as well as prepare more examples to be shown to users.

We present the details of our WSD method in Section 2. Method of example sentences expansion is shown in Section 3. Evaluation on WSD and expansion of examples is reported in Section 4. We discuss related work in Section 5. Finally we conclude the paper in Section 6.

2 Proposed WSD Method

The proposed method consists of two WSD classifiers: one is an example based classifier, the other is Robinson classifier.

2.1 Example Based WSD

In this paper, word senses or meanings are defined according to the Japanese-English dictionary EDICT. We develop the WSD classifier that calculates similarity between the input Japanese sentence and example sentences from a dictionary. Then choose example sentence which is the most similar to the input sentence.

In EDICT, word definitions often contain example sentences in both Japanese and English. Figure 1 shows the sense definitions **S** and example sentences **E** for the Japanese noun “*hanashi*” (story or discussion). For example, let us consider the case where the word sense of “*hanashi*” is to be disambiguated in input Japanese sentence **I1**.

I1 Han'nin ga tsukamatta to iu **hanashi** wa mettani kikanai.
(Rarely hear a story that the culprit got caught.)

<p>hanashi</p> <p>S1: story, talk , conversation , speech, chat</p> <p>E1: Mō koreijō sono hanashi o watashi ni kika senaide kudasai. (Please let me not hear of that story any more.)</p> <p>S2: discussions, argument, negotiation</p> <p>E2: 3-Jikan giron shitaga, wareware wa hanashi ga matomaranakatta. (After 3 hours of discussion we got nowhere.)</p>

Fig. 1. Sense and Example Sentences of “*hanashi*”

The classifier measures the similarity between **I1** and the example sentences **E1** and **E2**. Among them, **E1** may have the highest similarity with **I1**. Therefore, the classifier selects **S1** (story) as the correct sense definition for the word “*hanashi*”.

In order to choose example sentence (E) which is most similar to input sentence (I), we build an example based classifier which measures overall similarity $sim(I, E)$ as a sum of collocation similarity $coll(I, E)$ and similarity calculated by comparing syntactic relations $syn(I, E)$. It chooses the sense associated with the example sentence whose overall similarity score $sim(I, E) = coll(I, E) + syn(I, E)$ is highest and doesn’t choose any sense if the overall score is less than a threshold T , because the classifier cannot find an example sentence similar enough. Rare cases, when two or more senses have same score, sense with highest number of examples is chosen. Two similarity measures $coll(I, E)$ and $syn(I, E)$ are explained next.

Collocation Similarity. $coll(I, E)$ refers to collocation similarity score based on match sequences of n-grams of sizes 4, 5 and 6 between sentences I and E . 4-grams are a sequence of 4 words including a target word from a sentence such as $w_{-3}w_{-2}w_{-1}w_0$, $w_{-2}w_{-1}w_0w_1$, $w_{-1}w_0w_1w_2$ and $w_0w_1w_2w_3$, where w_0 is the target word and w_{-1} and w_1 are previous and next words to the target word, respectively and so on. Sequences for 5-grams and 6-grams are defined in the same way. $coll(I, E)$ score by using n-grams is calculated as per Equation (1). Weights for n-grams are determined in ad-hoc manner.

$$coll(I, E) = \begin{cases} 1 & \text{if one of 6-grams is same} \\ 0.75 & \text{elif one of 5-grams is same} \\ 0.5 & \text{elif one of 4-grams is same} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Syntactic Similarity. $syn(I, E)$ refers to syntactic similarity between two sentences I and E , for which we exploited the Japanese dependency structure usually represented by the linguistic unit called *bunsetsu*, which is a chunk consisting one or more content words and zero or more functional words. We use the same input sentence **I1** as an example to show such dependency structure in Figure 2.

Each *bunsetsu* has one head which is represented by bold face, followed by a *case marker* such as *ga*, *wa*⁴ or other functional words. Each head *bunsetsu* is always placed to the right of its modifier and the dependencies do not cross each other. We obtain such Japanese dependency structure by using analyzer Cabocha⁵.

We calculate $syn(I, E)$ by comparing syntactic relations r extracted from *bunsetsu* dependencies as:

$$r = w_1 - rel - w_2$$

$$rel = \begin{cases} \textit{case marker} & \text{if case marker follows } w_1 \\ \textit{adnominal} & \text{elif } POS(w_2) = \textit{Noun} \\ \textit{adverbial} & \text{otherwise} \end{cases} \quad (2)$$

Where w_1 and w_2 are a head of modifier and modifiee *bunsetsus* respectively and rel is the relation type. In the classifier, not all but only relations where either w_1 or w_2 is a target word are extracted. r_1 and r_2 are the extracted relations for sentence **11** from its dependency structure shown in Figure **2**.

$$r_1 : tsukama -adnominal- \mathbf{hanashi}$$

$$r_2 : \mathbf{hanashi} \quad -wa- \quad kika$$

Head word *tsukama* (catch) of *bunsetsu* #2 directly modifies *bunsetsu* #3, where head is the target word “*hanashi*” (story). Further ahead, “*hanashi*” directly modifies *bunsetsu* #5, therefore head “*kika*” (hear) is extracted as w_2 in r_2 .

Next, $syn(I, E)$ is defined as follows.

$$syn(I, E) = \sum_{(r_i, r_e) \in R_I \times R_E} s_r(r_i, r_e) \quad (3)$$

$$s_r(r_i, r_e) = \begin{cases} s_w(r_i(w_1), r_e(w_1)) & \text{if } r_i(w_2) = r_e(w_2) = t \\ & \text{and } r_i(rel) = r_e(rel) \\ s_w(r_i(w_2), r_e(w_2)) & \text{if } r_i(w_1) = r_e(w_1) = t \\ & \text{and } r_i(rel) = r_e(rel) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$s_w(w_i, w_j) = \begin{cases} 1 & \text{if } w_i = w_j \\ \frac{x}{8} & \text{otherwise} \end{cases} \quad (5)$$

In Equation **(3)**, $syn(I, E)$ is the sum of similarity scores $s_r(r_i, r_e)$ obtained by comparing all relations r_i and r_e extracted from input and example sentence respectively. Equation **(4)** compares two relations of same relation type rel and whose respective target word t is of same dependency structure in both relations i.e. either modifier or modifiee. Finally similarity of such relations is calculated by semantic similarity between words $s_w(w_i, w_j)$ as Equation **(5)**. Here w_i and

⁴ *ga* and *wa* are nominative (NOM) and topic (TOP) case markers, respectively.

⁵ <http://code.google.com/p/cabocha/>

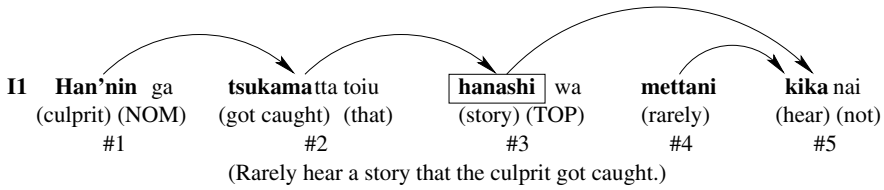


Fig. 2. Example of Bunsetsu Dependencies

w_j are modifier words from two relations that modifies the target word, vice versa are modifiee words when target word is the modifier. In Equation (5) x is the length of common prefix of semantic codes of two words in *Bunrui Goi Hyo* [5]. $s_w(w_i, w_j)$ is normalized to limit its score to < 1 [6].

Relations with Respect to Common Words. In calculation of $syn(I, E)$, only syntactic relations with respect to a target word are considered for the similarity between two sentences. It might be problematic because they seem insufficient to calculate sentence similarities precisely. To use more information for measuring similarity between sentences, we pay attention to common words in two sentences. For syntactic similarity, relations with respect to not only target word but also common words are used to obtain syntactic similarity. That is, in Equation (4), t refers to a target word or a common word. For example, there are two common words “*kika*” (hear) and “*hanashi*” (story) between **E1** and **I1**. A similarity between “*mettani* (rarely) - adverbial - *kika* (hear)” in **I1** and “*mo* (anymore) - adverbial - *kika* (hear)” in **E1** is also added to the score $syn(I1, E1)$. Considering common words to calculate $syn(I, E)$ will naturally affect in an increased recall, but may or may not affect the precision.

Hereafter, we call the example based WSD classifier which considers syntactic relations with respect to target word as RTW, while one considering relations with respect to common words as RCW. Note that both RTW and RCW also use collocation similarity $coll(I, E)$. We will empirically compare precision and recall of RTW and RCW in Section 4.

2.2 Robinson Classifier (ROB)

We incorporate a statistical approach of Bayesian classifier proposed by Robinson [6], popularly used in spam detection. We used the public tool [7] implementing the classifier as follows. For each sense s , the score S is calculated as Equation (6), then the sense which has the highest score is chosen. In Equation (7) and (8), P and Q estimate the likelihood and unlikelihood of a sense, respectively.

⁶ Note that a semantic code in *Bunrui Goi Hyo* is represented as 7 digits.

⁷ <http://sourceforge.net/projects/reverend/>

$$S = \frac{1 + (P + Q)/(P - Q)}{2} \quad (6)$$

$$P = 1 - ((1 - p(f_1)) \times (1 - p(f_2)) \times \dots \times (1 - p(f_n)))^{\frac{1}{n}} \quad (7)$$

$$Q = 1 - (p(f_1) \times p(f_2) \times \dots \times p(f_n))^{\frac{1}{n}} \quad (8)$$

$p(f_i)$ stands for the probability of each feature defined as:

$$p(f_i) = \frac{b(f_i)}{b(f_i) + g(f_i)} \quad i \in \{1, 2, \dots, n\} \quad (9)$$

where $b(f_i)$ and $g(f_i)$ in Equation (9) are the posterior probability $P(f_i|s)$ and $P(f_i|\bar{s})$ estimated by Maximum Likelihood estimation, respectively.

We use the conventional features constituting: Syntactic dependency relations (r in Equation (2)), collocation of bi-grams and tri-grams including the target word and bag-of-words of content words from each sentence. When no features from an input sentence occur in the training data, the most frequent sense in example database is chosen.

2.3 Combined Model

The final WSD classifier is an ensemble of example based and Robinson classifiers. First, precision oriented example based classifier is applied. When it does not choose a sense ($sim(I, E)$ is zero or less than a threshold), a sense from ROB is chosen. The main reason to combine example based classifier with Robinson classifier is to compensate recall because ROB is more robust. Note that the combined model can always choose a sense for given sentences.

3 Expansion of Example Sentences

In this section we describe the extraction of reliable example sentences. Since our reading assistant system will show examples in both Japanese and English, the goal here is to extract pairs of Japanese and English sentences. We used JENAAD which is an automatically sentence aligned parallel corpus and constitutes of 150,000 Japanese-English sentence pairs [7]. Further we used BerkeleyAligner⁸ and Morpha [8] to produce word alignments and lemmatized forms of words.

For each sense of the target word t , pairs of example sentences are extracted if they fulfill the following requirements:

- The Japanese sentence contains the target word t .
- There must exist an English word t_e aligned with t . t_e or a compound word including t_e should match against one of the words or compound words in the sense definition.
 - E.g., for target word “*deru*” with sense S_1 : {to go out, to exit, to leave}. If t_e is “exit” or “leave”, a sentence pair is extracted for the sense S_1 . If t_e is “go” and the succeeding word is “out” in English sentence, a sentence pair is also extracted.

⁸ <http://code.google.com/p/berkeleyaligner/>

- t_e should match against a word in sense definition for only one sense.
 - E.g., for target word “*tsukuru*” with sense S_1 : {to prepare, to brew} and S_2 : {to prepare, to make out, to write}, sentences are not extracted if t_e is “prepare”, since the sense of the target word is ambiguous.
- When a definition also consists of a short description in parenthesis, one of the content words in parenthesis should be contained in English sentence.
 - E.g., let us consider the target word “*dasu*” and its sense S_6 : {to produce (a sound)}. Sentence pair is extracted if t_e is “produce” and “sound” appears in an English sentence.

Above constraints are likely to reject many candidates, but it is crucial to extract sentences with high accuracy. Although pairs of Japanese and English sentences are added to example database, we only use Japanese examples for both example based and Robinson classifiers. The accuracy, coverage and effects on WSD upon expansion are discussed in the following section.

4 Evaluation

4.1 Data

To evaluate the performance of WSD, we prepared two sense tagged corpora, a development and an evaluation data. We first built the development data (D_d hereafter) to design our example based WSD method and optimize a threshold (T). It consists of 330 input sentences of 17 target words (8 nouns, 8 verbs and 1 adjective). Then, another sense tagged data is built as the evaluation data (D_e) to measure performance of our proposed method. It consists of 937 input sentences of 49 target words (23 nouns, 24 verbs and 2 adjectives). 17 target words on D_d are also target words on D_e . In both data, input sentences were excerpted from Mainichi Shimbun 1994 articles. Test sentences in D_d and D_e are mutually exclusive even for common target words. The correct senses are manually tagged by authors.

4.2 Results on Expansion of Example Sentences

Table [1](#) shows the statistics before and after expansion of example sentences, where T_d and T_e are sets of target words (TWs) on D_d and D_e , respectively. Statistics below the label E+ represents the figures after expansion including original numbers from EDICT. Number of example sentences are increased by about 1.5 times by expansion. Automatically expanded examples covered 36% of senses for T_e . Furthermore, number of senses with no example (6th column) are decreased. Note that senses with no example are crucial for our WSD method since such a sense is never chosen.

Among 5,470 expanded sentences for T_e , we randomly chose 10 sentences at most for each sense, then manually evaluate if extracted sentences are correct or not. Accuracy of automatically expanded example sentences was 85% as shown

Table 1. Comparison of Statistics Before and After Expansion (E+)

	# of TWs	Avg. Sense per TW	Total # of Eg Sents		Avg. Eg Sent per Sense		# of Senses with no Eg Sents	
			E+		E+		E+	
T_d	17	3.41	4,252	7,763	73.31	133.81	10	08
T_e	49	4.65	10,998	16,468	48.23	72.23	70	65

Table 2. Results on Examples Expansion

# of Sents	Correct	Incorrect	Accuracy
652	553	99	85%

in Table 2. Constraints on checking information in both languages effectively prunes false candidates generated due to misalignments or errors on morphological analysis. Among incorrect, 5 instances were due to wrong morphological analysis on Japanese while wrong sense is chosen in 94 instances.

On error analysis we found that one English word could correspond to two or more senses of target words in most cases where wrong sense is chosen. For example, S_1 : {inside, in} is one of senses of noun “*naka*”. S_1 means that a word is used with the name of a container or place to say where something is, such as “*tsukue no naka*” (in the desk). However, senses other than S_1 are often translated as “in”, such as “*jiyû no naka*” (in freedom). Another example is the noun “*hito*”. Two senses of this target word are S_1 : {man, person} and S_2 : {mankind, people}. If the target word of S_1 is translated as plural form of “person”, i.e. “people”, sentences are wrongly extracted for the sense S_2 . It is rather difficult to distinguish senses based on sense definitions in such cases, which tend to happen if differences among senses are subtle.

4.3 Results on WSD Classification

Table 3 reveals that the precision P, recall R and F-measure F of two example based classifiers RTW and RCW as well as precision of ROB, baseline BL and two combined models RTW+ROB and RCW+ROB on the development data. Here the baseline is the system which always selects the sense which has the highest number of example sentences. If more than one senses have same number of example sentences, it randomly chooses a sense. Since ROB, BL, RTW+ROB and RCW+ROB always choose a sense, not precision and recall but accuracy (ratio of agreement between gold sense and system’s sense) is shown for these systems.

As expected, when the threshold (T) is set high, precision of RTW and RCW increases but recall is dropped. Combination of precision oriented example based method with a robust ROB classifier is effective to improve the performance of WSD, since the accuracy of combined model outperforms both F-measure of RTW (or RCW) and accuracy of ROB.

Comparing RTW and RCW, RTW is better than RCW for precision and vice versa for recall and F-measure. When they are combined with Robinson classifier,

Table 3. Results on Development Data D_d

T	RTW			RCW			RTW	RCW
	P	R	F	P	R	F	+ROB	+ROB
0.0	0.72	0.48	0.58	0.66	0.53	0.59	0.67	0.66
0.3	0.76	0.35	0.48	0.68	0.45	0.54	0.67	0.66
0.6	0.83	0.26	0.39	0.73	0.36	0.48	0.66	0.66
0.9	0.90	0.16	0.28	0.79	0.29	0.42	0.64	0.66
E+								
0.0	0.70	0.57	0.63	0.62	0.59	0.60	0.67	0.61
0.3	0.71	0.55	0.62	0.63	0.55	0.58	0.67	0.61
0.6	0.74	0.45	0.56	0.67	0.49	0.56	0.65	0.60
0.9	0.77	0.36	0.49	0.68	0.40	0.51	0.62	0.59

ROB	
	0.62
E+	0.60
BL	
	0.59
E+	0.61

Table 4. Results on Evaluation Data D_e

T	RTW			RCW			RTW	RCW
	P	R	F	P	R	F	+ROB	+ROB
0.0	0.64	0.44	0.52	0.60	0.49	0.53	0.57	0.57
0.3	0.66	0.32	0.43	0.64	0.42	0.51	0.56	0.57
0.6	0.73	0.22	0.34	0.67	0.33	0.44	0.55	0.56
0.9	0.81	0.12	0.21	0.71	0.24	0.36	0.55	0.56
E+								
0.0	0.66	0.56	0.62	0.65	0.60	0.63	0.65	0.64
0.3	0.68	0.52	0.59	0.66	0.55	0.60	0.65	0.63
0.6	0.70	0.43	0.53	0.68	0.46	0.55	0.65	0.63
0.9	0.77	0.30	0.44	0.72	0.37	0.49	0.64	0.62

ROB	
	0.54
E+	0.60
BL	
	0.51
E+	0.58

RTW+ROB is better than RCW+ROB. This is because more precision oriented classifier RTW is preferable for combination with Robinson classifier.

By expanding example sentences, recall and F-measure are improved for both RTW and RCW, while precision is comparable when two systems are compared at same recall. For example, recall of RTW at T=0.3 and RCW at T=0.9 is around 0.35, while precision of RTW and RCW are 0.76 and 0.77, respectively. Example sentence expansion seems not contribute to a gain in the precision, although sentences are expanded with a high accuracy (85%). But it is not sure whether expansion has a positive impact on precision because the development data only consists of 17 target words. For combined models, RTW+ROB is comparable after expansion, while RCW+ROB and ROB are worse. Regardless of recall improvements, there is a drop in precision of RTW and RCW at same thresholds caused by expansion, which negatively influences the performance of combined models, especially RCW+ROB.

Considering optimization of the threshold, T=0 seems the best parameter for both RTW+ROB and RCW+ROB, although the accuracy does not highly depend on T in the combined model.

Table 4 shows results on the evaluation data D_e . Unlike results on D_d , the expansion of example database gives remarkable impacts for all classifiers on D_e . Especially, not only recall and F-measure but also precision of RTW and RCW is improved by expansion. Since D_e consists of more target words and test instances than D_d , results on D_e might be more reliable than D_d . Thus expansion of examples from parallel corpus is effective for WSD. The performance of RTW+ROB is better than RCW+ROB, but the difference is not so great compared with results on D_d . At the optimized threshold $T=0$, RTW+ROB achieves 0.65 accuracy, which is also the best on D_e .

Discussions. The performance of ROB is not so improved from BL on both with and without expanded examples. One of the reasons may be that example sentences in EDICT are used as training data. Especially, distribution of appearance of senses, which is known as effective statistics for WSD, can be trained from a sense tagged corpus, but not from example sentences in the dictionary, since it is not guaranteed that the numbers of examples for senses follow the real distribution. Further it makes more difficult to obtain sense distribution from automatically extracted sentences from the parallel corpus, since not all but only reliable sentences are extracted. In order to check the performance of other machine learning algorithm, we trained Support Vector Machine (SVM) and standard Naive Bayes classifiers from the same training data, but the performances were worse than the baseline. A collection of example sentences in a dictionary or automatically extracted examples seems less appropriate for supervised learning than a sense tagged corpus.

At the moment we set threshold (T) by looking at precision from the development data (i.e. a group of target words). We found that setting same T for all target words doesn't ensure an overall high precision. High precision is delivered from majority of target words at $T=0$, while a few require setting higher T. The way to optimize T for each target word should be investigated in future.

5 Related Work

As discussed in Section 1, our WSD can be regarded as a task of translation selection. Many researches in translation selection have been devoted. Dagan et al. proposed a method to use word co-occurrence in target language corpus [9]. Later approaches adopting co-occurrence statistics use simple mapping information between a source and its target words. Lee et al. showed the defect of using 'word-to-word' translation and proposed a translation selection method based on the 'word-to-sense' and 'sense-to-word' [10].

While example based Japanese WSD has also been studied. For example, Fujii et al. proposed a method for verb sense disambiguation, which measures sentence similarity based on semantic similarity of case filler nouns and weights of cases considering the influence of the case for WSD [11]. Then they proposed a method of selective sampling to reduce the cost of sense tagging to construct an example database. Target words are restricted to verbs in their research,

while WSD of nouns and adjectives is also considered in this paper. Shirai et al. proposed a method to disambiguate a sense of a word in a given sentence by finding the most similar example sentence in monolingual dictionary [12]. However, their method used only syntactic relations for measuring the similarity between sentences, while our method also considered collocation⁹ including the target word.

Approaches for automatic expansion of labeled example sentences have been seen in recent years. Fujita et al. expanded the labeled data by collecting sentences that include an exact match for example sentences in Iwanami dictionary [13]. Note that the extracted sentences would be much longer than ones in the dictionary, providing more information for WSD. Sentences extracted by Fujita’s method are homogeneous since only sentences similar to examples in the dictionary are obtained. While, our method can retrieve heterogeneous or wide variety of example sentences, which would be more suitable for WSD.

Melo et al. used a parallel corpus and an aligned sense inventory with sense tagged corpus to extract sense disambiguated example sentences [14]. In which sense of the target word is disambiguated by looking at the information in both language pairs individually. This approach is able to cover senses with no prior examples. However, disambiguation relies on an aligned sense inventory and a sense tagged corpora. Furthermore they proposed an algorithm, which chooses a set of valuable example sentences to showcase to user, by employing a weighing scheme using n-grams. However, they did not use extracted example sentences for WSD.

6 Conclusion

In this paper, we proposed a precision oriented example based WSD method. Proposed sentence similarity measures compute a score by exploiting collocation information and comparing syntactic dependency relations for a target word, just by using example sentences from an MRD. We also showed the reliability of these measures towards increasing precision by constraining a threshold. Being a precision oriented approach, robustness to the system comes by combining with a Robinson classifier. Reliable bilingual example sentences are extracted from an automatically aligned parallel corpus to enlarge the example database. Injection of extracted examples substantially increases the performance of all classifiers. Our method does not require sense tagged corpora. It achieved 65% accuracy, which is 7% better than the baseline.

As discussed in Subsection 4.3, it is rather difficult to infer sense distribution from examples in a MRD or automatically extracted examples. In future, it is important to guess the distribution without a sense tagged corpus to improve the performance of WSD. Currently, although we have a collection of pairs of Japanese and English sentences as example database, only Japanese sentences

⁹ To evaluate the contribution of collocation feature, we implemented RTW without collocation score $coll(I, E)$. Comparing RTW in Table 4 (T=0, with expanded example sentences), its precision was the same but recall was 4% lower.

are used for WSD. Another future work is to utilize information derived from English sentences for WSD.

References

1. Carpuat, M., Wu, D.: Improving statistical machine translation using word sense disambiguation. In: *The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 61–72 (2007)
2. Chan, Y.S., Ng, H.T.: Word sense disambiguation improves statistical machine translation. In: *45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, Prague, Czech Republic, pp. 33–40 (2007)
3. Navigli, R.: Word sense disambiguation: A Survey. *ACM Computing Surveys* 41(2), 1–69 (2009)
4. Okumura, M., Shirai, K., Komiya, K., Yokono, H.: SemEval-2010 task: Japanese WSD. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 69–74 (2010)
5. Natural Institute for Japanese Language and Linguistics (ed.): *Bunrui Goi Hyo. Dainippon Tosho* (2004)
6. Robinson, G.: A statistical approach to the spam problem. *Linux J.* 2003(107) (March 2003)
7. Utiyama, M., Isahara, H.: Reliable measures for aligning Japanese-English news articles and sentences. In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, ACL 2003*, vol. 1, pp. 72–79 (2003)
8. Minnen, G., Carroll, J., Pearce, D.: Robust, applied morphological generation. In: *Proceedings of the First International Natural Language Generation Conference*, 201–208 (2000)
9. Dagan, I., Itai, A.: Word sense disambiguation using a second language monolingual corpus. *Computational Linguistics* 20(4), 563–596 (1994)
10. Lee, H.A., Kim, G.C.: Translation selection through source word sense disambiguation and target word selection. In: *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, vol. 1, pp. 1–7 (2002)
11. Fujii, A., Inui, K., Tokunaga, T., Tanaka, H.: Selective sampling for example-based word sense disambiguation. *Computational Linguistics* 24(4), 573–597 (1998)
12. Shirai, K., Tamagaki, T.: Word Sense Disambiguation Using Heterogeneous Language Resources. In: Su, K.-Y., Tsujii, J., Lee, J.-H., Kwong, O.Y. (eds.) *IJCNLP 2004. LNCS (LNAI)*, vol. 3248, pp. 377–385. Springer, Heidelberg (2005)
13. Fujita, S., Fujino, A.: Word Sense Disambiguation by Combining Labeled Data Expansion and Semi-Supervised Learning Method. In: *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 676–685 (2011)
14. de Melo, G., Weikum, G.: Extracting sense-disambiguated example sentences from parallel corpora. In: *Proceedings of the 1st Workshop on Definition Extraction, WDE 2009*, pp. 40–46 (2009)

A Study on Hierarchical Table of Indexes for Multi-documents

Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu

Japan Advanced Institute of Science and Technology
{tho.le,nguyenml,shimazu}@jaist.ac.jp

Abstract. As a representation for multi-documents summarization, a table of indexes in hierarchical structure helps the readers understanding the content and the structure in semantics aspects. It also provides a navigation for the readers to quickly refer to interested information. In this paper, we introduce a framework to generate a hierarchical table of indexes. In which, we apply unsupervised clustering algorithm to create the hierarchical structure, and graph-based ranking method to extract keyphrases and form the indexes. The preliminary result is provided as the illustration for our approach.

Keywords: hierarchical summary, table of indexes, keyphrase extraction, clustering, unsupervised, graph based ranking.

1 Introduction

A summary of a document or a collection of documents is a condense representation of main ideas of the content. It is obvious that the summary of documents will help the readers gain the general ideas of documents. In case of the output of the summary for multiple documents, even if the summary is much shorter and more concise than the original documents, it is still difficult for reader to understand all main ideas in structural organization.

A form of summary that visualize the structure of document is *hierarchical summarization*, which has been noticed more than 10 years ago by work of Lawrie et al. [5] [6]. But the output are single words only [5], that may not express the ideas naturally. Also, the semantic aspects such as the *overlapping* or *supplementing* in the content of documents [6], especially in case of multiple documents, are still need more study. So, in this work, we represent the output of hierarchical summary in form of a combination of words, called *keyphrases*, which load the important information of the document. We also consider the semantic aspect of the content when extracting them from multiple documents.

So far, Branavan et al. [1] and Nguyen et al. [10] have used supervised technique to generate a tree wherein a node represents a segment of text and a title that summarizes its content with assumption that the hierarchical tree of summary is available. In this paper, we also focus on the constructing of hierarchical tree while trying to generate the summary of multiple documents. In details, we

create a structural summary in representation of a *hierarchical table of indexes* with unsupervised approach.

A hierarchical table of indexes (HTI) is similar to a table of indexes appearing at the back of books. Beside the role as a representation the summary, it helps readers quickly refer to their interested sections containing their interested key words."Hierarchical" in HTI means the set of indexes in lower tier contain more specific information than the higher one. The specific problem statement of this research is: given a collection of related documents in a specific field, our target is generating a hierarchical table of indexes. To solve that problem, the process involves to three steps:

- (i) Segment all documents into separate segments based on topics.
- (ii) Construct hierarchical tree of segments (HTS).
- (iii) Extract key-phrases from each segments, and generate HTI from HTS.

The first step is assumed to be available by apply existing methods of segmentation such as TextSeg [9], MinCutSeg [7], BayesSeg [3] in which a document is separated into segments based on topic by finding the maximum-probability segmentation. We will mainly focus on the second and third step of the process. We constructed the hierarchical structure of segments by unsupervised clustering approach; and generate HTI from HTS by extract keyphrases (terms) using graph rank based algorithm and sentence dependencies.

The construction of HTS is described in Section 2. The extracting key-phrases to generate HTI is described in Section 3. The experiment and preliminary result are illustrated in Section 4. Conclusions are given in Section 5, respectively.

2 Construct Hierarchical Table of Segments (HTS)

Let consider the set of segments obtained from first step to be a set of data points. A hierarchical tree of segments (HTS) is constructed by modelling the set of data points to a graph, and dividing segments into clusters. The graph is modelled as a triple of $G = (V, E, W)$, where $V = \{v_1, \dots, v_N\}$ is set of vertices, each vertex represent a segment; $E \subseteq V \times V$ is set of edges; $W = (w_{ij})_{i,j=1,\dots,N}$ is adjacency matrix, where each element w_{ij} is the weight of edge between two vertices v_i and v_j and $w_{ij} \in [0, 1]$. The weight of edge means how related between two data points, or the semantic similarity of two segments, the more related of two segments the more higher similarity it will be.

The HTS is constructed by dividing the set of data points of graph into clusters based on their semantic distances; then, each cluster will be re-divided into smaller pieces until it cannot be splitted; the sub-clusters obtained will form lower tier of hierarchical tree structure as described in details in Algorithm 1. At each tier, a threshold θ is applied to cut off the edge weights smaller than it, so that it will make the ideas in lower tier become more distinguishable than the higher one. In details, the completed graph at initial tier 0 is divided into some clusters; then, graphs corresponding to clusters are formed in which edges' weight lower than θ are removed. Threshold θ at child node is larger than its ancestor by

Algorithm 1. ConstructHTS

```

input : Graph  $G = (V, E, W)$ , Threshold  $\theta$ 
output: Hierarchical Tree of Segments

1 Clustering graph  $G$  to get a set of clusters  $C$ ;
2 if number of clusters in  $C > 1$  then
3    $\theta_k = \theta + \theta \times \gamma$ ; // constant  $\gamma$  is increasing coefficient
4   foreach cluster  $C_k \in C$  do
5     Construct sub-graph  $G_k = (V_k, E_k, W_k)$ , where  $V_k \in C_k$  and
6      $e_{ij} \in E_k \mid i, j \in V_k, w_{ij} \geq \theta_k$ ;
7     ConstructHTS( $G_k, \theta_k$ );
8   end

```

adding a percentage $\gamma \in [0, 1]$ to its ancestor's threshold. The recursive process will be stopped when the graph cannot be divided into clusters. As the result, the depth and the width of HTS are drawn automatically. In output, the root node of HTS includes branch node(s), the branch node contains other branch nodes or leaf nodes, and the leaf node includes segment(s).

3 Generate Hierarchical Table of Indexes (HTI)

The hierarchical table of indexes (HTI) will be built agglomerative from HTS, using graph ranking algorithm [3] to extract keyphrases from segments of leaf nodes, and consider them as indexes for corresponding segments. The keyphrases of the a segment in this context means a word or a group of words standing together and containing the significant information of the segment. To extract keyphrases from segment, the segment is modelled as a graph $G = (V, E)$. In which, V is set of vertices which are words from segment text; and E is the relation between vertices determining by sliding a co-occurrence window in size of N on the text, there will be a relation between two vertices if they occur in the window. When get the graph of candidate words, compute the vertex weights in graph with following formula until convergence [2]:

$$WS(V_i) = (1 - d) + d \times \sum_{V_j \in In(V_i)} \frac{WS(V_j)}{\sum_{V_k \in Out(V_j)} w_{jk}}$$

The outline of generating a HTI is described in Algorithm 2. The keyphrases for each segment are extracted by combining top ranked key words with the dependencies of words in sentence.

To extract keyphrases of a leaf node which includes many segments, all graphs modelled from segments are combined, then compute the vertex weight of combined graph, after that extract top ranked vertices as keyphrases for node. Keyphrases for branch nodes or root node are also extracted by similar way with the graph combined from children's graph. The nearer a node to root the

Algorithm 2. ConstructHTI

```

input : Hierarchical Tree of Segment HTS  $T$ 
output: Hierarchical Table of Indexes HTI

1 if  $T$  does not contain child node then // tree node  $T$  is leaf node
2   foreach segment  $S \in T$  do
3     Construct graph  $G = (V, E)$ , where  $V = \{\text{words} \in S\}$ ,  $E = \{e_{ij} \mid i, j \in V$ 
4       and  $i, j$  appear in co-occurrence window};
5     Compute all vertices' weight until convergence;
6     Rank graph vertices by descending order of vertices' weight;
7     From  $S$  extract keyphrases  $K = \{\text{combination of words appear at } p\% \text{ top}$ 
8       ranked in graph with sentence dependencies};
9     Construct new graph  $G' = (V', E')$ , where  $V' = V + K$  and
10       $E' = \{e_{ij} \mid i \in V, j \in K \text{ and } \exists w \in j, w = i\}$ ;
11   end
12 else // tree node  $T$  is root or branch node
13   From graphs  $(G_i = (V_i, E_i))_{i=1, \dots, N}$  constructed from the children of  $T$ ,
14   construct graph  $H = (V_h, E_h)$ , where  $V_h = \{V_1 \cup \dots \cup V_N\}$ ,
15    $E_h = \{E_1 \cup \dots \cup E_N \cup E\}$  with  $E = \{e_{ij} \mid \exists w \in j \in E_p, \exists v \in j \in E_q, w = v\}$ ;
16 end
17 Compute all vertices' weight until convergence;
18 Rank all graph vertices again;
19 Extract top ranked vertices to be keyphrases for the current node;

```

more general of keyphrases will be, and the more nearer to leaf node the more specific of keyphrases of the node is.

4 Experiment

Experiment data is Pension Law in both Japanese and English version collected from <http://www.japaneselawtranslation.go.jp/>. We illustrate the proposed approach on a document named *Act on Controls on the Illicit Export and Import and other matters of Cultural Property* in category of *Education and Culture*.

In preprocessing step, the title and heading of documents are omitted. At first step, all documents are then be segmented by TextSeg [9]. In the second step, we model complete graph $G = (V, E, W)$ from set of segments, with edge weight is text similarity computed in semantic aspect as described in [2]. Next, we apply Algorithm 1 with an existing unsupervised clustering algorithm Affinity Propagation [4] to construct HTS with initial threshold $\theta = 0.5$ and increase coefficient $\gamma = 50\%$. In the last step, keyphrases are extracted from the segments of leaf nodes of HTS to generate HTI using Algorithm 2.

The experiment is run in both English and Japanese version of law. A piece of the result is illustrated in the Fig. 1, where the leftmost box is the root containing the highest ranked indexes for all, it is decomposed into branches with more specific indexes added into lower tier. Table 1 describes the result

and manual evaluation on the generating of HTI for the given document. In each language, the total number of keyphrases is the number of all keyphrases in HTI generated by proposed approach. The HTI is then showed to human and get the respond of which keyphrase in HTI is acceptable or it is important to the main ideas of the text. Because of the characteristics of language, the phrase in English and Japanese content different kind of part-of-speeches, it causes an approximately 5% different between the rate of English and Japanese.



Fig. 1. The illustration of output for law in Japanese and English

Table 1. Result of keyphrases extraction

Language	Total number of keyphrases	Keyphrases accepted by human	Accepted Rate
English	150	68	45.3%
Japanese	96	39	40.6%

5 Conclusions

In this work, we introduced a framework to generate hierarchical table of indexes for multiple documents. In which, we proposed an approach to construct hierarchical tree of segments using existing unsupervised clustering algorithm with the depth and wide of the hierarchical structure is drawn automatically. By grouping similar segments into the same clusters based on semantic distance, we argue that segments with *overlapped* content will be grouped in the same cluster. In addition, when the higher tier in hierarchical structure represent the keyphrases of all sub-clusters in lower tier, it also represent that the lower tiers contain more details information and can be considered as *supplement* for the content in higher tiers. We also proposed an approach to extract keyphrases by combining sentence dependencies and graph rank based method algorithm, and then generate hierarchical table of indexes from the tree of segments.

In future work, we plan to apply statistical model to constructing the hierarchical structure and combine the supportive knowledge into extracting keyphrases while generating the table of indexes.

References

1. Branavan, S.R.K., Deshpande, P., Barzilay, R.: Generating a table-of-contents. In: Proc. of ACL 2007, Prague, Czech Republic, pp. 544–551 (June 2007)
2. Corley, C., Mihalcea, R.: Measuring the semantic similarity of texts. In: Proc. of EMSEE 2005, Stroudsburg, PA, USA, pp. 13–18 (2005)
3. Eisenstein, J., Barzilay, R.: Bayesian unsupervised topic segmentation. In: Proc. of the EMNLP 2008, Stroudsburg, PA, USA, pp. 334–343 (2008)
4. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. *Science* 315, 972–976 (2007)
5. Lawrie, D., Bruce Croft, W., Rosenberg, A.: Finding topic words for hierarchical summarization. In: Proc. of ACM SIGIR 2001, pp. 349–357. ACM, New York (2001)
6. Lawrie, D.J., Bruce Croft, W.: Generating hierarchical summaries for web searches. In: Proc. of ACM SIGIR 2003, pp. 457–458. ACM, New York (2003)
7. Malioutov, I., Barzilay, R.: Minimum cut model for spoken lecture segmentation. In: Proc. of CoLing/ACL 2006, Sydney, Australia, pp. 25–32 (July 2006)
8. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: EMNLP, pp. 404–411 (2004)
9. Utiyama, M., Isahara, H.: A statistical model for domain-independent text segmentation. In: Proc. of ACL 2001, Stroudsburg, PA, USA, pp. 499–506 (2001)
10. Cuong, N.V., Le Minh, N., Akira, S.: Learning to generate a table-of-contents with supportive knowledge. *IEICE Transactions on Information and Systems*, 423–431 (March 2011)

Finding Good Initial Cluster Center by Using Maximum Average Distance

Samuel Sangkon Lee^{1,*} and Chia Y. Han²

¹ Dept. of Computer Science and Engineering, Jeonju University, South Korea
samuel@jj.ac.kr

² School of Computing Sciences and Informatics, University of Cincinnati, OH, USA
han@ucmail.uc.edu

Abstract. This paper presents an improved algorithm for partitioning clustering used for large data. The proposed method is based on the K-means algorithm which is commonly used because of easy implementation and easy control of time complexity when there are 'N' number of documents, compared to other systems. Analysis on the performance of clustering experiments with an actual document data set, the proposed method obtained a performance of 8.8% higher than the randomly selected clustering in terms of center values. Furthermore, consistent clustering results were obtained by taking care of the dependency of clustering results on initial centers.

Keywords: Initial Center Selection, Maximum Average Distance, Time Complexity, K-Means Algorithm, Document Clustering.

1 Introduction

Since entering the modern era of Internet, documents that are produced by all the Internet users have been growing exponentially. In the past, website pages and blog documents were mostly found on established sites, but now, thanks to the development social networking services (SNS), various types of documents, like in Facebook or Twitter, have been produced and distributed in great volume. As a result, it takes more time for users to get the information they want. Thus, search engines that would help users find the exact information they want quickly have become more critical. One of the key requirements for this kind of search service consists of document clustering which reveals similarities among the created documents.

Clustering methods can be divided into two types: hierarchical clustering [1][9] and partitioning cluster [6][10]. In processing a massive amount of data, partitioning clustering methods are more efficient. The K-means algorithm, which is one of the partitioning clustering methods, is widely used. It offers several important aspects in terms of handling mass data of today's digital world, because the K-means-based algorithms are easy to implement and can handle data

* Corresponding author.

relatively quickly with $O(n)$ of time complexity provided 'n' is the number of documents. However, the performance of the K-means algorithms is greatly dependent on how the center of the initial cluster is set. If the initial cluster center is not well chosen, the results can be poor. Therefore, an important step in the K-means algorithm is to repeat the center allocation and recalculation to move the center to the right position. If the initial center of the cluster is leaned to a particular position, however, the allocation-recalculation frequency can dramatically increase, or improper clustering results may take place. This study aims to improve the performance of the K-means algorithm effectively by selecting the center of rational initial clustering through calculation instead of using the conventional random sampling-based initial clustering center selection method. For this, the distance among initial cluster centers is maximized. Then, the cluster centers are evenly distributed in the data set. Using the method proposed in this paper, the center of the evenly distributed initial clusters can produce far more accurate results of document clustering, compared to the initial centers which were randomly selected. The method proposed in this study requires additional time for selection of the initial cluster centers. However, it has attempted to prove that the total clustering time could be reduced by decreasing allocation-recalculation frequency through a test.

In chapter 2, previous studies on clustering techniques and K-means algorithm are stated. In chapter 3, the average maximum distance-based technique is proposed as a way to select initial centers. In chapter 4, a system in which the proposed clustering technique was actually applied to document clustering is constructed, and a test is conducted using a test data set. Then, the results are analyzed and evaluated. In chapter 5, conclusion and future studies are mentioned.

2 Related Studies

2.1 K-Means Algorithm

The K-Means algorithm is the most commonly used partitioning clustering. The concept of this algorithm is to minimize the average Euclidean distance between the documents and the center of the document clustering. If the center of the clustering is the mean or centroid of the document, it can be defined as follows:

In the Equation (1), S is a set of cluster documents while c is a particular document belonging to the cluster. The documents are expressed in vector. In K-means algorithm, the cluster can be considered as a sphere which has centroid just like the center of gravity.

2.2 Initial Value

The performance of K-Means algorithm greatly varies depending on how initial centers are selected. According to previous studies, initial centers consist of 'k' randomly selected documents or 'k' random coordinates within a set of documents. When the results of the document clustering were examined, this method

the mean or centroid ($\bar{\mu}$),

ω is a set of cluster documents,

\bar{x} is a particular document belonging to the cluster.

$$\bar{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\bar{x} \in \omega} \bar{x} \tag{1}$$

$c_i^{initial}$ is the i th cluster vector.

d_j refers to j th document vector.

$$c_i^{initial} = avg\ big\left(\sum_{j=1}^3 d_j\right) \tag{2}$$

is bound to a large variation. To solve this problem, there have been a lot of studies on initial center setting. In her study [11], Shinwon discovered that the characteristics of initial center of cluster belong to a particular set of documents with a common attribute. The center vector of initial cluster has been set by selecting three documents which are expressed in index term and weighted value in the selected initial cluster instead of selecting one random document. The triple center setting algorithm can be stated as follows:

This study has attempted to select initial centers using relatively diverse properties, but this method has also unable to overcome the limitations of random selection completely.

Rafail et al.[8] has proposed the separation conditions to find 'k' initial centers which are situated very close to the optimum centers, considering the distance that is the division size of each cluster and the fact that each optimum center can have an initial center. In this algorithm, the process to get initial center selection by [Rafail et al. (2006)] is shown below:

1. Select x and y based on the probability proportional to $\|x - y\|^2$ and set them to c_1 and c_2 respectively. Here, $x, y \in X$ can be obtained, whereas X is a set of total data.
2. Using more than 2 ($i \geq 2$) conventional centers (c_1, \dots, c_i), the probability ($\min_{j \in \{1, 2, \dots, i\}} \|x - c_j\|^2$) proportional to the random data $x \in X$ is calculated. Then, it is set to c_{i+1} .
3. Repeat the process above (2.) until i reaches k .

Paul and Rafail [7] used K-Means algorithm for communication protocol in order to apply it to a communication security system. It is called 'two-party K-Means clustering protocol.' For implementation of this protocol, it is necessary to come up with algorithm for initial center setting to get a single data set. It is shown in initial center selection algorithm Paul and Rafail (2007) below: the basic concept is to locate the center, starting from the center in order to find the initial centers in the whole documents. In other words, initial center (or seed) is gathered in the center of the whole documents, and the distribution-recalculation of K-means algorithm is repeated. Here, this paper has attempted

to describe the difference between the method proposed by Paul and Rafail and the one mentioned in this study. In the method introduced by Paul and Rafail, clustering begins after collecting initial centers in the middle. In the method proposed in this study, initial centers are allocated to the outside of set of the whole document as much as possible, and the best centers are discovered and clustering through the distribution-recalculation process.

1. Calculation of the median of whole document: $C = \frac{1}{n} \left(\sum_{i=1}^n D_i \right) \epsilon$
2. Calculation of the distance between all data and the median: $\tilde{C}_i^0 = Dist^2(C, D_i) \epsilon$
3. Calculation of mean distance: $\bar{C} = \frac{1}{n} \left(\sum_{i=1}^n \tilde{C}_i^0 \right) \epsilon$
4. Selection of the first center: $\mu_1 = D_i, Pr[\mu_1 = D_i] = \frac{\bar{C} + \tilde{C}_i^0}{2n\bar{C}} \epsilon$
5. Iteration to select the rest centers: $\mu_j, j = 2, \dots, k \epsilon$
 - 5.1 $\tilde{C}_i^{j-1} = Dist^2(\mu_{j-1}, D_i), 1 \leq i \leq n \epsilon$
 - 5.2 $\tilde{C}_i^j = \min\{\tilde{C}_i^l\}_{l=0}^{j-1}, 1 \leq i \leq n \epsilon$
 - 5.3 $\bar{C} = average \tilde{C}_i^j (over all 1 \leq i \leq n) \epsilon$
 - 5.4 $\mu_j = D_i, Pr[\mu_j = D_i] = \frac{\tilde{C}_i^j}{n\bar{C}} \epsilon$

For example, let's say there are two virtual users (Bob and Alice) who share data for communication, and the data are DB and DA respectively. If it is assumed that the centers are respectively, this algorithm uses two data sets under the same method for security of the transmitted messages.

In terms of selection of initial centers based on certain calculation, the method proposed by Rafail et al. [8] and Paul and Rafail [7] is similar to the method mentioned in this study. However, initial centers are positioned too close to the center of the data set (a set of whole documents of clustering). Even though performance has slightly improved, compared to a simple random selection method, the effect to distribute initial centers is still minor. Therefore, this study has attempted to propose a new system which works well when the initial centers of data set are close to the median and even when they are away from the median as well.

3 Center Set Using Maximum Average Distance

3.1 Calculation of Distance

In this section, a method to distribute the initial cluster centers is introduced as an attempt to improve the conventional K-means algorithm. If the centers are distributed, it can be prevented for the randomly selected initial cluster centers

from being leaned to a particular area. In addition, clustering processing speed and the accuracy of clustering results can be enhanced. If a set of the initial cluster centers is 'C,' the algorithm can be defined as follows:

Whereas, c_i is the center of i th cluster, and c_{avg} is the mean from c_1 to c_k . In other words, a key point of the algorithm proposed in this study is to maximize the distance of the centers from c_1 to c_k . The below shows an algorithm used to get the initial cluster centers:

$$C = \max \sum_{i=1}^K \|c_{avg} - c_i\|^2 \tag{3}$$

1. Select 'K' centers randomly.
2. In terms of $x \in X$,
 - 2.1 Select the candidate which is the closest to x
 $candidate\ Cluster \leftarrow \min_{i=0, \dots, k} dist(x, c_i)$
 - 2.2 Substitute the current center with new candidate clusters and calculate new average distance as follows:
 $newDistAvg \leftarrow avg \sum_{i=1}^k |c_{avg} - c_i|^2$
if $c_i = candidate\ Cluster$ **then** $|c_{avg} - x|$
 - 2.3 If new distance average is greater than old distance average,
if $(newDistAvg > oldDistAvg)$ **then** $c_i \leftarrow x$
3. return $\{c_1, \dots, c_k\}$

The initial center selection method stated above can be explained again as follows: First, 'K' centers are randomly selected. After measuring the distance with the old centers against all x in a data set 'X' in Stage 2.1, the closest center (c_j) is selected. In Stage 2.2, the closest center is replaced with new x , and the average distance among the centers is calculated. In Stage 2.3, if the average distance when x is replaced with c_j is greater than the average distance among the current centers, the current center (c_j) is replaced with x . Figure 1 below shows the result of the simulation of the selection of initial cluster centers, which has been performed using 2D data:

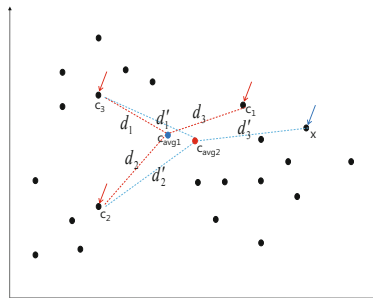


Fig. 1. Transition of Initial Centers with Application of the Method Proposed in This Study

As shown in Figure 4, if assumed that there already are three centers (c1, c2, c3), let's find the center which is the closest to the new data 'x.' When the distance between c1, c2, c3 and x was compared, it has been confirmed that c1 is the closest point. Then, the distance between each point and average (d1, d2, d3) is calculated after adding x instead of c1 as shown new distance average = $1/K \sum_{i=1}^k d_{prime j}$. Then, it is compared with the distance between current points (d1, d2, d3) as old distance average = $1 / K \sum_{i=1}^k d_j$.

When two average distances were compared, new distance average which was estimated by adding x instead of c1 was greater than old distance average. Therefore, x is substituted for c1. Therefore, x, c2, c3 are now compared to x as old distance average. This process is repeated against all x in the data set 'X.'

3.2 Time Complexity

Compared to the conventional center selection method, the new method proposed in this study requires a new process to calculate maximum average distance. This new process is the overhead of the algorithm proposed in this study. Considering this overhead, the time consumed for clustering is calculated as follows. The time which is additionally consumed during the new allocation and recalculation process for initial center transition can be estimated as shown in equation $T(\text{initial center setting}) + T(\text{allocation and recalculation})$.

As shown in the algorithm in Figure 3 above, a total of 5K time has been spent ((1K: time spent to select the point closest to x in Stage 2.1) + (1K: time spent to get average of the new point when the conventional center was replaced with 'x' in Stage 2.2) + (1K: time spent to calculate distance between the average and each point)). Here, let's examine time complexity in the conventional K-means algorithm. If time complexity is $O(kN)$, time complexity for calculation of maximum average distance is $O(5KN)$. Then, time for allocation and recalculation is estimated. For this, the time (1K) spent for allocation of each document to clustering work and the time (1K) spent to recalculate centers against the documents in each cluster are required. Therefore, the equation for allocation and recalculation shall be $O(2iKN)$. Whereas, i is the frequency until the allocation and recalculation process is completed. Therefore, the time spent for entire clustering is $O(5KN) + O(2iKN) = O(N)$. Because i and K are constants, the time spent for entire clustering is the linear function of $O(5KN) + O(2iKN)$.

In addition, because of the said reason, the time spent for selection of initial centers does not have a big impact on the time spent for entire clustering, which shall be verified through the experiment below.

4 New Clustering Method Based on the Proposed Algorithm

In this section, it is attempted to apply the improved K-means algorithm to the document clustering system. First of all, the overview of the entire system

is handled, and a clustering system has been implemented by using K-means algorithm which fits for application. The system has been configured as follows: First of all, all document sets go through the following two stages. The first stage is preprocessing. It consists of term weighting, normalization of weighted values and selection of particular features. The second stage is actual clustering which has been proposed in this study. In this process, initial cluster centers are selected first. Then, documents are clustered by K-means algorithm which begins from the selected initial centers.

4.1 Experimental Result

In terms of data for the experiment, 20 newsgroup data sets were used. A 20-newsgroup data is commonly used in applications of diverse machine learning systems such as document classification or document clustering. A total of 2,000 documents have been used. The test started from the randomly selected initial centers. After calculating maximum average distance, clustering was conducted using the new centers. In terms of document normalization, cosine normalization and pivot normalization have been used.

4.2 Analysis of Test Results

The documents in "sci.space" are accurate because they are relatively clearer than the documents in other mains in terms of document bounds. In wrong documents, wrong results have occurred due to the following words; 'widespread,' 'air,' 'cannibalism' and 'NASA.'

Table 1 above shows the result of clustering which began from the randomly selected centers while Table 2 reveals the test result of clustering which started from the new initial centers. To compare two tables, the results of F-measure are stated in Table 3.

It has been confirmed that performance improved by 8.8% when clustering was performed using the centers adjusted by the maximum average distance proposed in this study, compared to the case in which randomly selected centers were used. Table 4 below compares old algorithm with the center recalculation frequency after allocating each document to cluster. Figure 2 reveals the frequency in graphs.

As shown in the Figure 2, the amplitude of execution times was large in case of old algorithm (random), which means that difference in allocation and recalculation frequency is large depending on initial centers. In this system (maximum average distance), however, mostly even frequency was observed. In other words, the result of clustering is not-dependent on initial center setting. As shown in Table 4 below, in addition, average frequency decreased from 21.1 times in old algorithm to 14.8 times in new algorithm. Even though the time consumed for additional calculation to select initial centers was considered, the time spent for whole clustering decreased by about 0.7%. In summary, the new algorithm proposed in this study can improve accuracy by 8.8% and reduce clustering time by 0.7%.

C_i^0	$ c_i \text{ class} = j^0$															$f\text{-measure}(c_i)$					
1 ⁰	41.7 ⁰	16.3 ⁰	0 ⁰	1.8 ⁰	0 ⁰	1.8 ⁰	0 ⁰	0 ⁰	6.4 ⁰	4.8 ⁰	0 ⁰	1.2 ⁰	0 ⁰	0 ⁰	4.2 ⁰	0 ⁰	0 ⁰	17.8 ⁰	0.8 ⁰	3.2 ⁰	56.4% ⁰
2 ⁰	0 ⁰	51.5 ⁰	26.9 ⁰	2.2 ⁰	1.2 ⁰	0.8 ⁰	0 ⁰	0.8 ⁰	6 ⁰	0 ⁰	3.4 ⁰	1 ⁰	0 ⁰	2.2 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	3.2 ⁰	0.8 ⁰	34.8% ⁰
3 ⁰	0 ⁰	3 ⁰	41.6 ⁰	10 ⁰	18.2 ⁰	18 ⁰	0 ⁰	0 ⁰	0 ⁰	0.8 ⁰	3.2 ⁰	0 ⁰	0 ⁰	2.2 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	3 ⁰	40.4% ⁰
4 ⁰	0 ⁰	0 ⁰	21.2 ⁰	33.8 ⁰	18.6 ⁰	4.6 ⁰	5.6 ⁰	1 ⁰	2 ⁰	1.2 ⁰	2.2 ⁰	3.2 ⁰	1.2 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0.8 ⁰	3.8 ⁰	0.8 ⁰	40.6% ⁰
5 ⁰	0 ⁰	19.7 ⁰	11.8 ⁰	0.7 ⁰	46.8 ⁰	3.3 ⁰	0 ⁰	0 ⁰	2.2 ⁰	0.8 ⁰	2.2 ⁰	0.9 ⁰	3.3 ⁰	2.3 ⁰	2.3 ⁰	0 ⁰	0 ⁰	0 ⁰	1.2 ⁰	2.6 ⁰	43.3% ⁰
6 ⁰	0 ⁰	19.6 ⁰	0.8 ⁰	0 ⁰	13.5 ⁰	50.6 ⁰	2.2 ⁰	0 ⁰	5 ⁰	0 ⁰	2.3 ⁰	3.6 ⁰	0.9 ⁰	0 ⁰	0.8 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0.7 ⁰	50.3% ⁰
7 ⁰	0 ⁰	3.3 ⁰	0.9 ⁰	0 ⁰	14.6 ⁰	1.6 ⁰	59.8 ⁰	0.7 ⁰	0 ⁰	4.4 ⁰	1.2 ⁰	5.4 ⁰	2.6 ⁰	4.8 ⁰	0 ⁰	0.7 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	71.1% ⁰
8 ⁰	0 ⁰	1.4 ⁰	0 ⁰	0.9 ⁰	0 ⁰	1.2 ⁰	0 ⁰	44.6 ⁰	2.8 ⁰	0 ⁰	7.6 ⁰	6.6 ⁰	2.6 ⁰	19.9 ⁰	3.8 ⁰	0 ⁰	0 ⁰	0 ⁰	0.3 ⁰	8.3 ⁰	60.2% ⁰
9 ⁰	0 ⁰	11.4 ⁰	0 ⁰	0.9 ⁰	0 ⁰	4.4 ⁰	0 ⁰	0 ⁰	42.4 ⁰	0 ⁰	5.2 ⁰	3 ⁰	0.2 ⁰	27.3 ⁰	0.9 ⁰	1.2 ⁰	0 ⁰	0 ⁰	2.3 ⁰	0.8 ⁰	45.7% ⁰
10 ⁰	0 ⁰	0.9 ⁰	0 ⁰	2.3 ⁰	0 ⁰	2.6 ⁰	0 ⁰	0 ⁰	45.3 ⁰	6.3 ⁰	7.2 ⁰	2.2 ⁰	2.6 ⁰	0.9 ⁰	0 ⁰	0.8 ⁰	21.5 ⁰	6.2 ⁰	1.2 ⁰	0 ⁰	54.9% ⁰
11 ⁰	0 ⁰	16.7 ⁰	0.3 ⁰	2.2 ⁰	0 ⁰	0.3 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	73.8 ⁰	3.6 ⁰	0.9 ⁰	0 ⁰	2.2 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	62.0% ⁰
12 ⁰	0 ⁰	0.9 ⁰	0 ⁰	0 ⁰	0.3 ⁰	0 ⁰	0 ⁰	0 ⁰	6.6 ⁰	0 ⁰	4.3 ⁰	60.6 ⁰	0 ⁰	2.4 ⁰	2.4 ⁰	0 ⁰	5.2 ⁰	0 ⁰	16.5 ⁰	0.8 ⁰	56.7% ⁰
13 ⁰	0 ⁰	7.6 ⁰	2.6 ⁰	0.7 ⁰	2.8 ⁰	0 ⁰	0.7 ⁰	0 ⁰	3.6 ⁰	0 ⁰	7.6 ⁰	3.2 ⁰	52.3 ⁰	10.8 ⁰	4.8 ⁰	0 ⁰	2.4 ⁰	0 ⁰	0 ⁰	0.9 ⁰	61.6% ⁰
14 ⁰	0 ⁰	10.4 ⁰	0 ⁰	0.7 ⁰	0 ⁰	1.8 ⁰	0 ⁰	0 ⁰	3.6 ⁰	0 ⁰	6.6 ⁰	2.8 ⁰	0 ⁰	58.9 ⁰	4.2 ⁰	0 ⁰	1.8 ⁰	0 ⁰	9.2 ⁰	0 ⁰	48.6% ⁰
15 ⁰	0 ⁰	14.1 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	1 ⁰	0 ⁰	0.7 ⁰	4.2 ⁰	2.2 ⁰	0 ⁰	5.2 ⁰	69.2 ⁰	0 ⁰	0.9 ⁰	0 ⁰	0.7 ⁰	1.8 ⁰	69.8% ⁰
16 ⁰	0 ⁰	1.6 ⁰	0 ⁰	5.6 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0.8 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	91.1 ⁰	0 ⁰	0.9 ⁰	0 ⁰	0 ⁰	0 ⁰	94.0% ⁰
17 ⁰	0 ⁰	10.2 ⁰	0 ⁰	0.5 ⁰	0 ⁰	2.2 ⁰	0 ⁰	0 ⁰	1 ⁰	2.6 ⁰	0.7 ⁰	2.8 ⁰	1.8 ⁰	3 ⁰	0.9 ⁰	0 ⁰	66.3 ⁰	0.7 ⁰	2.2 ⁰	5.1 ⁰	60.4% ⁰
18 ⁰	0 ⁰	6.6 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	5.2 ⁰	2.8 ⁰	0.9 ⁰	0 ⁰	0 ⁰	0 ⁰	16.2 ⁰	60.7 ⁰	0 ⁰	7.6 ⁰	52.8% ⁰
19 ⁰	1 ⁰	0.8 ⁰	0 ⁰	0 ⁰	0 ⁰	4.2 ⁰	0 ⁰	0 ⁰	1 ⁰	0 ⁰	1.2 ⁰	3.8 ⁰	0 ⁰	1 ⁰	0.9 ⁰	0.9 ⁰	16.2 ⁰	4.2 ⁰	61.7 ⁰	3.1 ⁰	58.0% ⁰
20 ⁰	5.2 ⁰	0 ⁰	0 ⁰	4.2 ⁰	0 ⁰	3.8 ⁰	0 ⁰	0 ⁰	2.2 ⁰	4.3 ⁰	0.7 ⁰	0 ⁰	0.8 ⁰	0 ⁰	0.7 ⁰	0 ⁰	12.3 ⁰	21 ⁰	3.8 ⁰	41 ⁰	45.3% ⁰

Fig. 2. Result of Pivot Normalization and Application of Random Centers

C_i^0	$ c_i \text{ class} = j^0$															$f\text{-measure}(c_i)$					
1 ⁰	41.4 ⁰	0.7 ⁰	9.8 ⁰	0.9 ⁰	4.2 ⁰	4.3 ⁰	16.7 ⁰	6.2 ⁰	0 ⁰	0 ⁰	0 ⁰	2.4 ⁰	0 ⁰	1.4 ⁰	0 ⁰	2.2 ⁰	0 ⁰	0 ⁰	9.8 ⁰	57.1% ⁰	
2 ⁰	0 ⁰	45.4 ⁰	20.7 ⁰	13.2 ⁰	5.3 ⁰	10.8 ⁰	0 ⁰	3.2 ⁰	0 ⁰	0 ⁰	1.2 ⁰	0 ⁰	0 ⁰	0.6 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	43.6% ⁰
3 ⁰	0 ⁰	8.5 ⁰	51.3 ⁰	12.3 ⁰	20.9 ⁰	3.2 ⁰	0 ⁰	0 ⁰	1.2 ⁰	0.7 ⁰	1.4 ⁰	0 ⁰	0.4 ⁰	0.1 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	46.5% ⁰
4 ⁰	0 ⁰	8.1 ⁰	15.7 ⁰	48.6 ⁰	4.6 ⁰	6.8 ⁰	0 ⁰	2.2 ⁰	1.2 ⁰	0 ⁰	0 ⁰	1.8 ⁰	0 ⁰	10.6 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0.4 ⁰	0 ⁰	47.4% ⁰
5 ⁰	0 ⁰	12.3 ⁰	9.3 ⁰	5.2 ⁰	47.9 ⁰	3.2 ⁰	0 ⁰	3.2 ⁰	0 ⁰	0 ⁰	0 ⁰	1.6 ⁰	15.5 ⁰	1.8 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	45.5% ⁰
6 ⁰	0 ⁰	9.6 ⁰	13.8 ⁰	5.9 ⁰	7.3 ⁰	53.3 ⁰	0 ⁰	6.2 ⁰	0.9 ⁰	0.7 ⁰	0 ⁰	0 ⁰	1.6 ⁰	0.7 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	53.4% ⁰
7 ⁰	0 ⁰	1.4 ⁰	0 ⁰	9.2 ⁰	0.7 ⁰	0 ⁰	81.6 ⁰	0 ⁰	3.2 ⁰	0 ⁰	1.2 ⁰	0 ⁰	1.8 ⁰	0 ⁰	0.9 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	77.8% ⁰
8 ⁰	0.2 ⁰	0 ⁰	0 ⁰	0.7 ⁰	0.9 ⁰	0.9 ⁰	0 ⁰	58.7 ⁰	32.4 ⁰	0 ⁰	1.2 ⁰	0 ⁰	1.8 ⁰	1 ⁰	0 ⁰	2.2 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	57.8% ⁰
9 ⁰	0 ⁰	0.6 ⁰	0 ⁰	0 ⁰	0.9 ⁰	0.7 ⁰	0 ⁰	4.2 ⁰	76.9 ⁰	0 ⁰	0.9 ⁰	2.8 ⁰	0 ⁰	9.4 ⁰	0.8 ⁰	1.2 ⁰	0 ⁰	1.6 ⁰	0 ⁰	0 ⁰	64.2% ⁰
10 ⁰	0 ⁰	0 ⁰	0 ⁰	1.8 ⁰	0 ⁰	0 ⁰	0 ⁰	4.2 ⁰	53.7 ⁰	30.7 ⁰	0.1 ⁰	0.9 ⁰	0 ⁰	1.2 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	7.4 ⁰	0 ⁰	64.7% ⁰
11 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0.9 ⁰	4.9 ⁰	0 ⁰	2.4 ⁰	3.2 ⁰	69.7 ⁰	0 ⁰	0.4 ⁰	16.7 ⁰	1.8 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	65.2% ⁰
12 ⁰	0.9 ⁰	1.9 ⁰	0 ⁰	0 ⁰	2.6 ⁰	2.2 ⁰	7.2 ⁰	0.9 ⁰	1.8 ⁰	0 ⁰	68.9 ⁰	0 ⁰	10.8 ⁰	1 ⁰	0 ⁰	0 ⁰	0 ⁰	1.8 ⁰	0 ⁰	0 ⁰	77.0% ⁰
13 ⁰	0.9 ⁰	10.6 ⁰	0 ⁰	2.2 ⁰	0 ⁰	3.8 ⁰	0.8 ⁰	4.2 ⁰	9.8 ⁰	3.6 ⁰	1.8 ⁰	1.2 ⁰	56.1 ⁰	0 ⁰	2.8 ⁰	0 ⁰	1.2 ⁰	0.9 ⁰	0.1 ⁰	0 ⁰	66.4% ⁰
14 ⁰	0 ⁰	2 ⁰	0 ⁰	0.7 ⁰	0.9 ⁰	0.7 ⁰	2.6 ⁰	1.2 ⁰	3 ⁰	0 ⁰	2.8 ⁰	0.9 ⁰	0 ⁰	82.9 ⁰	1.6 ⁰	0 ⁰	0.7 ⁰	0 ⁰	0 ⁰	0 ⁰	56.9% ⁰
15 ⁰	0 ⁰	2.2 ⁰	0 ⁰	0 ⁰	0.9 ⁰	1.6 ⁰	0 ⁰	0 ⁰	0.7 ⁰	0.8 ⁰	0 ⁰	0 ⁰	13.4 ⁰	75.8 ⁰	0 ⁰	0 ⁰	4.6 ⁰	0 ⁰	0 ⁰	0 ⁰	78.8% ⁰
16 ⁰	0.9 ⁰	1 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0.8 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	96.1 ⁰	0 ⁰	1.2 ⁰	0 ⁰	0 ⁰	0 ⁰	0 ⁰	93.3% ⁰
17 ⁰	0 ⁰	0 ⁰	0 ⁰	0.9 ⁰	2.2 ⁰	0.7 ⁰	3.8 ⁰	1.2 ⁰	0.7 ⁰	0 ⁰	0 ⁰	0.8 ⁰	0 ⁰	1.8 ⁰	0 ⁰	6.7 ⁰	76 ⁰	0 ⁰	5.2 ⁰	0 ⁰	76.2% ⁰
18 ⁰	0.7 ⁰	0 ⁰	0 ⁰	0 ⁰	4.6 ⁰	0.8 ⁰	0 ⁰	0 ⁰	0.7 ⁰	2.2 ⁰	0.9 ⁰	0 ⁰	11.1 ⁰	0 ⁰	0 ⁰	0 ⁰	78.1 ⁰	0.9 ⁰	0 ⁰	0 ⁰	80.1% ⁰
19 ⁰	0 ⁰	1.2 ⁰	0 ⁰	1.2 ⁰	3.6 ⁰	0 ⁰	0.6 ⁰	0.9 ⁰	2.2 ⁰	0.8 ⁰	0 ⁰	0 ⁰	0.9 ⁰	8.8 ⁰	1.9 ⁰	1.2 ⁰	0 ⁰	12.7 ⁰	64 ⁰	0 ⁰	71.5% ⁰
20 ⁰	0 ⁰	2.4 ⁰	0 ⁰	2.4 ⁰	5.6 ⁰	2.2 ⁰	0 ⁰	3.6 ⁰	0.7 ⁰	0 ⁰	0 ⁰	1.6 ⁰	4.8 ⁰	6.8 ⁰	0 ⁰	0 ⁰	19.4 ⁰	0 ⁰	0.8 ⁰	49.7 ⁰	59.4% ⁰

Fig. 3. Results of Pivot Normalization and New Initial Center Method

φ	$f\text{-measure}(c_i)$	average
Random	56.4 ⁰ 34.8 ⁰ 40.4 ⁰ 40.6 ⁰ 43.3 ⁰ 50.3 ⁰ 71.1 ⁰	

Test Frequency	1	2	3	4	5	6	7	8	9	10	average
Old Algorithm	20	24	24	14	21	20	18	26	21	23	21.1
New Algorithm	14	15	16	15	14	14	15	14	16	15	14.8

Fig. 5. Cluster Allocation Frequency

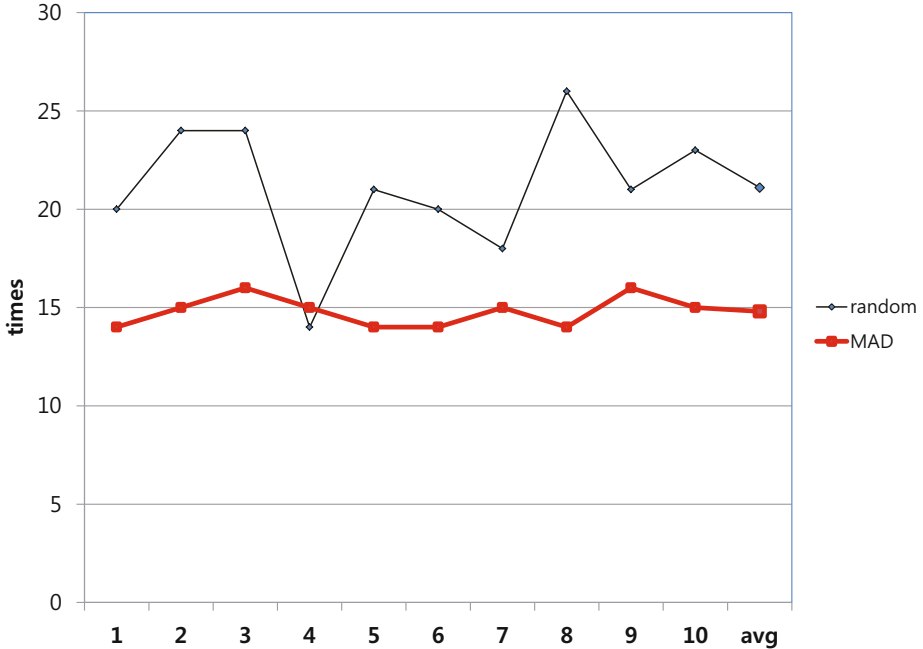


Fig. 6. Document Allocation Routine Frequency

5 Conclusions

This study has improved the performance of K-mean algorithm among partitioning clustering which is convenient in clustering large data. K-means is algorithm which is commonly used because of easy implementation and easy control of time complexity when there are 'N' number of documents, compared to other systems. However, bad results can occur depending on initial center setting.

Clustering accuracy can be enhanced by allocating centers as far as possible without randomly selecting clustering centers in the early clustering stage. This study has proposed K-means algorithm to get relatively consistent clustering

results. According to an analysis on the performance of clustering after applying it to an actual document data set, the method proposed in this study was higher than the randomly selected clustering by about 8.8% in terms of center values. In addition, even though it was expected that additional time would be necessary because of selection of initial centers, time complexity ($O(N)$) got linearly with the number of documents, and the time spent for whole clustering was rather reduced by decreasing allocation-center recalculation frequency after allocating the documents to each cluster. Furthermore, consistent clustering results were obtained by taking care of the dependency of clustering results on initial centers.

Clustering has been widely used in many industrial sectors such as information search, email clustering, communication protocol clustering and medical information clustering. The K-means algorithm which is enhanced based on maximum average distance in this study can be applied to these sectors. However, a further study needs to be performed to make it applicable to hierarchical clustering as well as to partitioning clustering.

References

1. Adami, G., Avesani, P., Sona, D.: Clustering Documents in a Web Directory. In: Proceedings of the 5th ACM International Workshop on Web Information and Data Management, pp. 66–73 (2003)
2. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, pp. 331–338. Cambridge University Press (2008)
3. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall Advanced Reference Series. Prentice-Hall, Inc., Upper Saddle River (1988)
4. Lloyd, S.P.: Least Squares Quantization in PCM. Special Issue on Quantization, IEEE Trans. Inform. Theory 28, 129–137 (1982)
5. McQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
6. Meedeniya, D.A., Perera, A.S.: Evaluation of Partition-Based Text Clustering Techniques to Categorize Indic Language Documents. In: IEEE International Advance Computing Conference (IACC 2009), pp. 1497–1500 (2009)
7. Bunn, P., Ostrovsky, R.: Secure Two-Party k-Means Clustering. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, pp. 486–497 (2007)
8. Ostrovsky, R., Rabani, Y., Schulman, L.J., Swamy, C.: The Effectiveness of Lloyd-Type Methods for the K-Means Problem. In: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 165–176 (2006)
9. Sahoo, N., Callan, J., Krishnan, R., Duncan, G., Padman, R.: Incremental Hierarchical Clustering of Text Documents. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 357–366 (2006)

10. Yu, Y., Bai, W.: Text Clustering based on Term Weights Automatic Partition. In: 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), pp. 373–377 (2010)
11. Lee, S.: A Study on Hierarchical Clustering Using Advanced K-Means Algorithm for Information Retrieval. Doctoral Thesis, Chonbuk University, Jeonju, South Korea (2005) (in Korean)
12. Pavan, K.K., Rao, A.A., Rao, A.V.D., Sridhar, G.R.: Single Pass Seed Selection Algorithm for K-Means. *Journal of Computer Science* 6(1), 60–66 (2010)

Applying a Burst Model to Detect Bursty Topics in a Topic Model

Yusuke Takahashi¹, Takehito Utsuro¹, Masaharu Yoshioka², Noriko Kando³,
Tomohiro Fukuhara⁴, Hiroshi Nakagawa⁵, and Yoji Kiyota⁶

¹ University of Tsukuba, Tsukuba, 305-8573, Japan

² Hokkaido University, Sapporo, 060-0808, Japan

³ National Institute of Informatics, Tokyo 101-8430, Japan

⁴ National Institute of Advanced Industrial Science and Technology,
Tokyo 135-0064, Japan

⁵ University of Tokyo, Tokyo 113-0033, Japan

⁶ NEXT Co., Ltd., Tokyo, 108-0075, Japan

Abstract. This paper focuses on two types of modeling of information flow in news stream, namely, burst analysis and topic modeling. First, when one wants to detect a kind of topics that are paid much more attention than usual, it is usually necessary for him/her to carefully watch every article in news stream at every moment. In such a situation, it is well known in the field of time series analysis that Kleinberg's modeling of bursts is quite effective in detecting burst of keywords. Second, topic models such as LDA (latent Dirichlet allocation) are also quite effective in estimating distribution of topics over a document collection such as articles in news stream. However, Kleinberg's modeling of bursts is usually applied only to bursts of keywords but not to those of topics. Considering this fact, we propose how to apply Kleinberg's modeling of bursts to topics estimated by a topic model such as LDA and DTM (dynamic topic model).

Keywords: Time Series News, Topic Model, Kleinberg's Burst Model.

1 Introduction

This paper studies two types of modeling of information flow in news stream, namely, burst analysis and topic modeling. Both types of modeling, to some extent, aim at aggregating information and reducing redundancy within the information flow in news stream.

First, when one wants to detect a kind of topics that are paid much more attention than usual, it is usually necessary for him/her to carefully watch every article in news stream at every moment. In such a situation, it is well known in the field of time series analysis that Kleinberg's modeling of bursts [1] is quite effective in detecting burst of keywords.

Second, topic models such as LDA (latent Dirichlet allocation) [2] and DTM (dynamic topic model) [3] are also quite effective in estimating distribution of

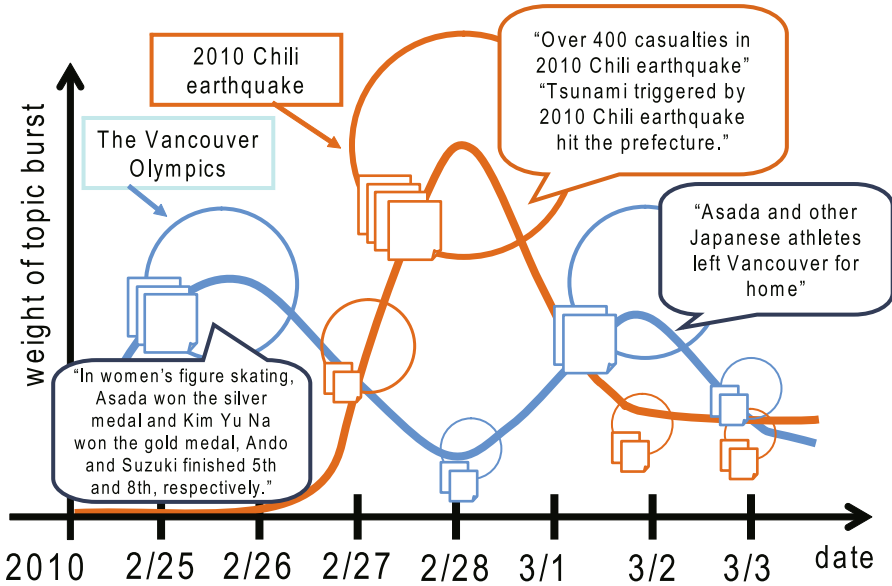


Fig. 1. Bursty Topics in Time Series News Stream

topics over a document collection such as articles in news stream. Unlike LDA, in DTM, we suppose that the data is divided by time slice, for example by date. DTM models the documents (such as articles of news stream) of each slice with a K -component topic model, where the k -th topic at slice t smoothly evolves from the k -th topic at slice $t - 1$.

Based on those arguments above, this paper proposes how to integrate the two types of modeling information flow in news stream. Here, it is important to note that Kleinberg’s modeling of bursts is usually applied only to bursts of keywords but not to those of topics. Thus, we propose how to apply Kleinberg’s modeling of bursts to topics estimated by a topic model such as DTM. Typical results of applying the proposed technique to time series news stream can be illustrated as in Figure 1. In this example, after we estimate time series topics through DTM, we can detect bursty topics such as “the Vancouver Olympics” and “2010 Chile earthquake” by the proposed technique.

2 Kleinberg’s Bursts Modeling

[1] proposed two types of frameworks for modeling bursts. The first type of modeling is based on considering a sequence of message arrival times, where a sequence of messages is regarded as bursty if their inter-arrival gaps are too small than usual. The second type of modeling is, on the other hand, based on the case where documents arrive in discrete *batches* and in each batch of documents, some are *relevant* (e.g., news text contains a particular word) and some are *irrelevant*.

In this second type of bursts modeling, a sequence of batched arrivals could be considered bursty if the fraction of relevant documents alternates between reasonably long periods in which the fraction is small and other periods in which it is large. Out of the two modelings, this paper employs the latter, which is named as *enumerating bursts* in [11].

2.1 Enumerating Bursts

Suppose that there are m batches of documents; the t -th batch B_t in the sequence $\mathbf{B} = (B_1, \dots, B_m)$ of m batches contains r_t relevant documents out of a total of d_t . Let $R = \sum_{t=1}^m r_t$ and $D = \sum_{t=1}^m d_t$. Now, we define a 2-state automaton \mathcal{A}^2 , where the state q_0 denotes the non-burst state, while the state q_1 denotes the burst state. For each q_i of the two states q_0 and q_1 , there is an expected fraction p_i of relevant documents. Set $p_0 = R/D$, and $p_1 = p_0s$, where $s > 1$ is a scaling parameter, while $p_1 \leq 1$ holds for p_1 . In this paper, we set s as 2.

Viewed in a generative fashion, state q_i produces a mixture of relevant and irrelevant documents according to a binomial distribution with probability p_i . The cost of a state sequence $\mathbf{q} = (q_{i_1}, \dots, q_{i_m})$ in \mathcal{A}^2 is defined as follows. If the automaton is in state q_i when the t -th batch B_t arrives, a cost of

$$\sigma(i, r_t, d_t) = -\ln \left[\binom{d_t}{r_t} p_i^{r_t} (1 - p_i)^{d_t - r_t} \right]$$

is incurred, since this is the negative logarithm of the probability that r_t relevant documents would be generated using a binomial distribution with probability p_i . There is also a cost of $\tau(i_t, i_{t+1})$ associated with the state transition from q_{i_t} to $q_{i_{t+1}}$. $\tau(i_t, i_{t+1})$ is defined so that the cost of moving from the non-burst state to the burst state is non-zero, but there is no cost for the automaton to end a burst and drop down to a non-burst. Specifically, when $j > i$, moving from q_i to q_j incurs a cost of $(j - i)\gamma$, where $\gamma > 0$ is a parameter;¹ and when $j \leq i$, the cost is 0.

$$\tau(i, j) = \begin{cases} (j - i)\gamma & (j > i) \\ 0 & (j \leq i) \end{cases}$$

In this paper, we set γ as 1.

Then, given a sequence of batches $\mathbf{B} = (B_1, \dots, B_m)$, the goal is to find a state sequence $\mathbf{q} = (q_{i_1}, \dots, q_{i_m})$ that minimizes the cost function:

$$c(\mathbf{q} | \mathbf{B}) = \left(\sum_{t=0}^{m-1} \tau(i_t, i_{t+1}) \right) + \left(\sum_{t=1}^m \sigma(i_t, r_t, d_t) \right)$$

¹ In [11], $\tau(i, j)$ is defined not as $(j - i)\gamma$, but as $(j - i)\gamma \ln m$, where m is the number of batches in the sequence $\mathbf{B} = (B_1, \dots, B_m)$. In this paper, we omit the term $\ln m$ in this definition for simplicity.

2.2 Weight of a Keyword Burst

Given an optimal state sequence, bursts of positive intensity correspond to intervals in which the state is q_1 rather than q_0 . For such a burst $[t_k, t_l]$, we can define the *weight* of the burst to be:

$$bw(t_k, t_l) = \sum_{t=t_k}^{t_l} (\sigma(0, r_t, d_t) - \sigma(1, r_t, d_t))$$

In other words, the weight is equal to the improvement in cost incurred by using state q_1 over the interval rather than state q_0 . Observe that in an optimal sequence, the weight of every burst is non-negative. Intuitively, then, bursts of larger weight correspond to more prominent periods of elevated activity.

In [1] as well as in this paper, this amount of the burst weight can be considered as that of a keyword if some of the documents in a batch are regarded as *relevant* when containing a particular keyword, while some are regarded as *irrelevant* when not containing a particular keyword. Throughout this paper, we referred to this notion of keyword bursts by simply regarding a document as *relevant* when containing a particular word.

Moreover, in this paper, we apply the burst model to news stream, where, as the time slice $t (t = 1, \dots, m)$ of the batch sequence $\mathbf{B} = (B_1, \dots, B_m)$, we use their dates. Here, we measure the weight of a keyword burst for each individual date. In such a case, we can assume $t_k = t_l (= t)$, and then, the weight of a keyword burst can be denoted as below:

$$bw(t) = bw(t, t)$$

3 Topic Model

As a time series topic model, this paper employs DTM (dynamic topic model) [3]. Unlike LDA (Latent Dirichlet Allocation) [2], in DTM, we suppose that the data is divided by time slice, for example by date. DTM models the documents (such as articles of news stream) of each slice with a K -component topic model, where the k -th topic at slice t smoothly evolves from the k -th topic at slice $t - 1$.

In this paper, in order to model time series news stream in terms of a time series topic model, we consider date as the time slice t . Given the number of topics K as well as time series sequence of batches each of which consists of documents represented by a sequence of words w , on each date t (i.e., time slice t), DTM estimated the distribution $p(w|z_n)$ ($w \in V$) of a word w given a topic z_n ($n = 1, \dots, K$) as well as that $p(z_n|b)$ ($n = 1, \dots, K$) of a topic z_n given a document b , where V is the set of words appearing in the whole document set. In this paper, we estimate the distributions $p(w|z_n)$ ($w \in V$) and $p(z_n|b)$ ($n = 1, \dots, K$) by a Blei's toolkit², where for the number of topics $K = 20$, as well as $\alpha = 0.01$.

²<http://www.cs.princeton.edu/~blei/topicmodeling.html>

4 Modeling Bursty Topics in a Topic Model

Based on the formalization of Kleinberg's bursts modeling presented in section 2, this section proposes how to model bursty topics among those estimated through the topic modeling framework of previous section.

In the modeling of keyword bursts, we simply regard a document as *relevant* when containing a particular keyword, and then count the number r_t of relevant documents out of a total of d_t . In the modeling of topic bursts, on the other hand, we first regard a document b as *relevant* to a certain topic z_n that are estimated through the DTM topic modeling procedure, to the degree of the amount of the probability $p(z_n|b)$. We then estimate the number r_t of relevant documents out of a total of d_t simply by summing up the probability $p(z_n|b)$ over the whole document set:

$$r_t = \sum_b p(z_n|b)$$

Once we have the number r_t , then we can estimate the total number of relevant documents throughout the whole batch sequence $\mathbf{B} = (B_1, \dots, B_m)$ as

$R = \sum_{t=1}^m r_t$. Having the total number of documents throughout the whole batch

sequence as $D = \sum_{t=1}^m d_t$, we can estimate the expected fraction of relevant doc-

uments as $p_0 = R/D$. Then, by simply following the formalization of keyword bursts presented in section 2, it is quite straightforward to model bursty topics in a topic model. The weight of a topic burst is also introduced through precisely the same formalization as the weight of a keyword burst.

5 Evaluation

5.1 Applying the Topic Model to News Stream

As the news stream documents set for evaluation, during the period from February 1st to March 31st, 2010, we collected 10,976 Yomiuri newspaper articles³, 9,210 Nikkei newspaper articles⁴, and 6,710 Asahi newspaper articles⁵ which amount to 29,896 newspaper articles in total. To those newspaper articles, the DTM topic modeling toolkit is applied and 20 topics are estimated for each date during the period from February 1st to March 31st, 2010. For each of the estimated 20 topics, Table 1 shows labels manually annotated to each topic as well as five keywords with the topmost probability $p(w|z_n)$ for each topic z_n (for the date March 1st, 2010). Here, with DTM as the topic model, the distribution $p(w|z_n)$ of a word w given a topic z_n ($n = 1, \dots, K$) may vary day by day.

³ <http://www.yomiuri.co.jp/>

⁴ <http://www.nikkei.com/>

⁵ <http://www.asahi.com/>

Table 1. 20 Topics estimated by DTM (for the date March 1st, 2010.)

manually annotated labels	five keywords w with the topmost probability $p(w z_n)$ for each topic z_n
economy	ドル (dollar), ユーロ (Euro), 上昇 (rise), 市場 (market), 動き (movement)
Toyota vehicle recalls	トヨタ (Toyota), リコール (recall), 問題 (problem), 社長 (President), 公聴会 (hearing)
the Vancouver Olympics	選手 (athlete), 女子 (women's), 日本 (Japan), バンクーバー (Vancouver), 3月1日 (March 1st)
natural phenomenon	津波 (tsunami), チリ (Chili), メートル (meter), 被害 (damage), 午後 (p.m.)
Japanese politics	首相 (prime minister), 政府 (government), 予算 (budget), 国会 (diet), 民主党 (Democratic Party of Japan)
Ichiro Ozawa's suspected illegal donations	民主党 (Democratic Party of Japan), 北教組 (the Hokkaido Prefectural Teachers' Association), 選挙 (election), 自民党 (the Liberal Democratic Party of Japan), 参院選 (the Lower House election)
foreign politics	中国 (China), 大統領 (president), 米国 (the United States), 日本 (Japan), 政府 (government)
business	社長 (president), 企業 (business), 販売 (sale), 会社 (company), 開発 (develop)
performance of companies	10, 月 (months), 発表 (announcement), 連続 (continuity), 価格 (price)
traffic	運転 (drive), 事故 (accident), キロ (kilometer), 合わせ ("awase", <i>substring of an idiom</i>), 午後 (p.m.)
trial	被告 (the accused), 判決 (court decision), 裁判員 (citizen judges), 裁判 (trial), 事件 (case)
sport, product information	ネット (Net), サイト (site), インターネット (Internet), 発売 (launching), 携帯電話 (cell phone)
the Futenma issue	知事 (prefectural governor), 問題 (issue), 政府 (government), 米軍 (US forces), 沖縄 (Okinawa)
entertainment	監督 (director), 映画 (movie), 作品 (production), 放送 (air), 舞台 (stage)
criminal case	容疑者 (suspect), 容疑 (charge), 逮捕 (arrest), 事件 (case), 女性 (woman)
local news	販売 (sale), 合わせ ("awase", <i>substring of an idiom</i>), イベント (event), 店舗 (shop), 商品 (product)
school, column	自分 (self), 子ども (children), 生徒 (student), 学校 (school), 参加 (participation)
society	対象 (candidate), 受け (receiving), 調査 (survey), 採用 (recruitment), 制度 (institution)
medical service	病院 (hospital), 患者 (patient), 受け (receiving), 医師 (doctor), 医療 (medical service)
local administration	年度 (year), 市長 (mayor), 地域 (region), 計画 (plan), 施設 (facility)

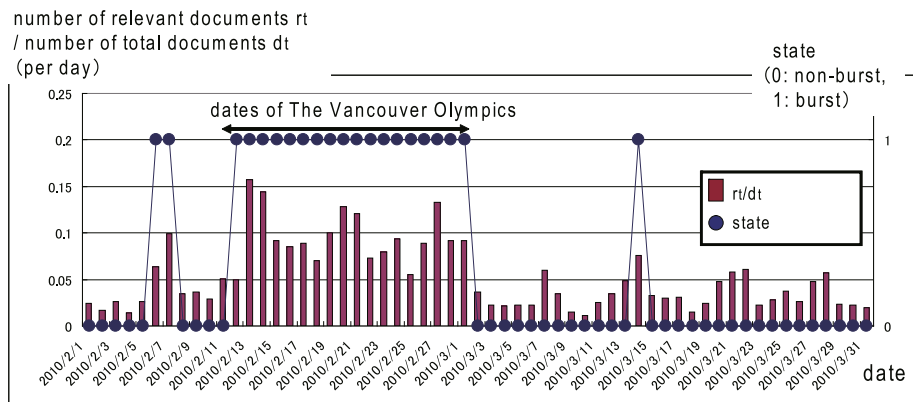
Table 2. Evaluation Results: Detecting Bursty Topics

# of detected bursts	# of correctly detected bursts	precision (%)
109	91	83.5

However, in the case of the evaluation data set presented in this paper, the five keywords with the topmost probability $p(w|z_n)$ for each topic z_n do not vary so much through the period from February 1st to March 31st, 2010.

5.2 Detecting Bursty Topics

As the results of evaluation on detecting bursty topics by the proposed method for the period from February 1st to March 31st, 2010, Table 2 shows the number of detected bursts, that of correctly detected bursts as well as the precision. The precision of detecting bursty topics is over 83%, which is reasonably high. In most of the falsely detected bursty topics, on the other hand, estimated topics themselves cover news articles of relatively broad issues⁶. For those topics, it is quite by chance for bursty topics to be detected, since they are detected simply because news articles covering broad ranges of issues are observed more than usual on some date. For example, on some date, the number of news articles on foreign politics are slightly more than usual, where they are all closely related to the topic “foreign politics”. In such a case, the proposed method detected the topic “foreign politics” as bursty on that date. In order to avoid such errors, it is quite promising to invent a technique of judging confidence as well as coherence of the estimated topics within the framework of DTM, and then to avoid detecting bursty topics when they have too little confidence and/or too little coherence.

**Fig. 2.** Optimal State Sequence for the Topic “the Vancouver Olympics”

⁶ Out of the total 20 topics, they are the following 10: “Japanese politics”, “foreign politics”, “performance of companies”, “traffic”, “trial”, “sport, product information”, “entertainment”, “criminal case”, “local news”, and “school, column”.

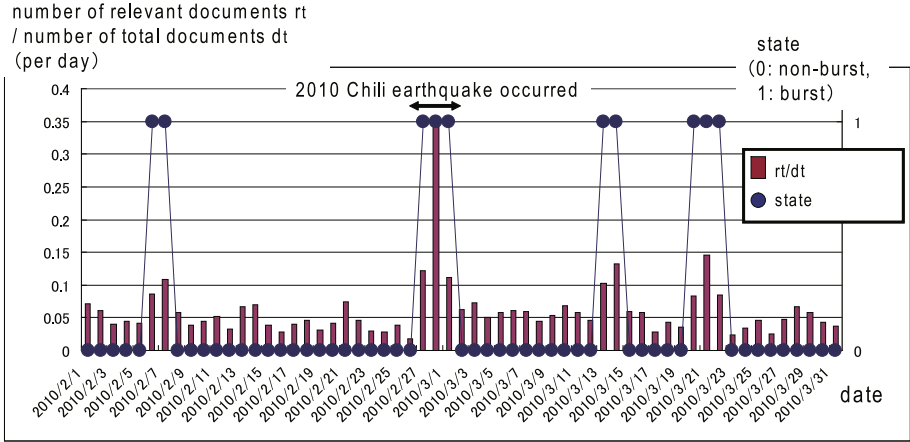


Fig. 3. Optimal State Sequence for the Topic “natural phenomenon”

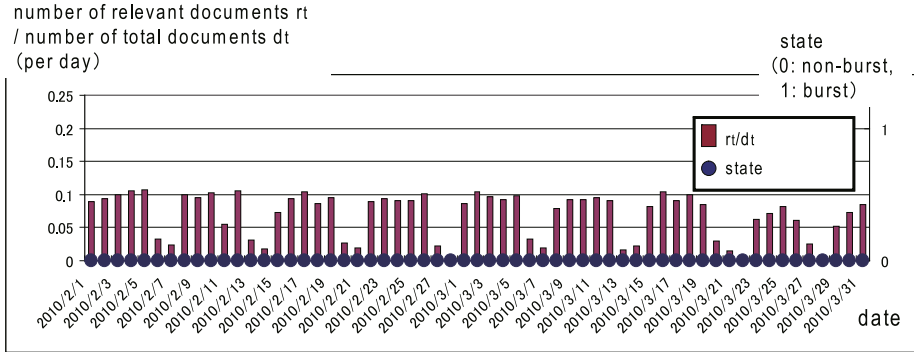


Fig. 4. Optimal State Sequence for the Topic “economy”

Figures 2 ~ 5 plot the optimal state sequence for the topics “the Vancouver Olympics”, “natural phenomenon”, “economy”, and “society”. Figure 2 clearly shows that the proposed method detects the bursty topic “the Vancouver Olympics” precisely during the dates of the Vancouver Olympic Games⁷. Figure 3 shows that the proposed method detects the bursty topic “natural phenomenon” when natural disasters such as earthquakes occurred. Figures 4 and 5, on the other hand, show that bursts are not detected for the topics such as “economy” and “society”, for which news are constantly reported everyday, and the rate r_t/d_t per day is mostly fixed within a certain range throughout the whole year.

⁷ The burst states preceding the dates of the Olympic games are simply due to the newspaper articles which report the training of the Japanese Olympic athletes. The burst state about two weeks after the Olympic closing day is also simply due to the newspaper articles which report the opening of the Vancouver Paralympic games.

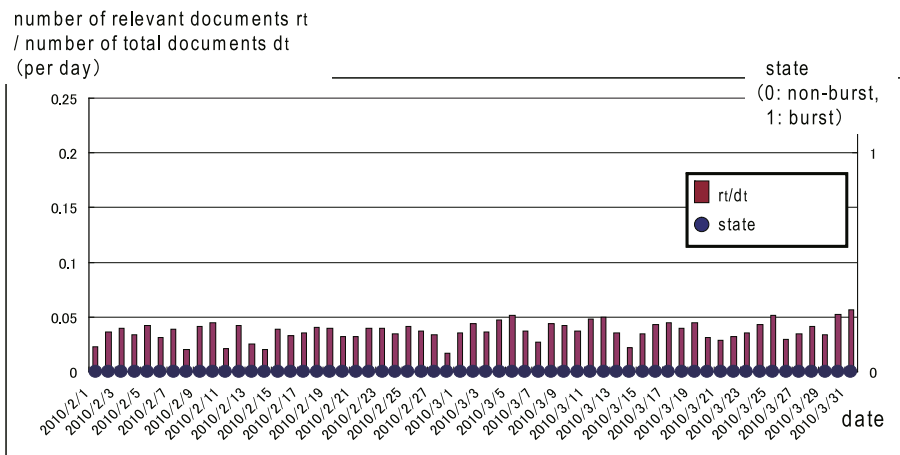


Fig. 5. Optimal State Sequence for the Topic “society”

5.3 Analysis on the Weight of a Topic Burst

Figure 6 plots changes in the weight of topic bursts for the topics “Toyota vehicle recalls”, “Ichiro Ozawa’s suspected illegal donations”, “the Vancouver Olympics”, and “natural phenomenon”, where we focus on the period from February 1st to March 3rd, 2010 as the dates for the plots. As can be seen from these plots, we claim that the weight of topic bursts is quite helpful for

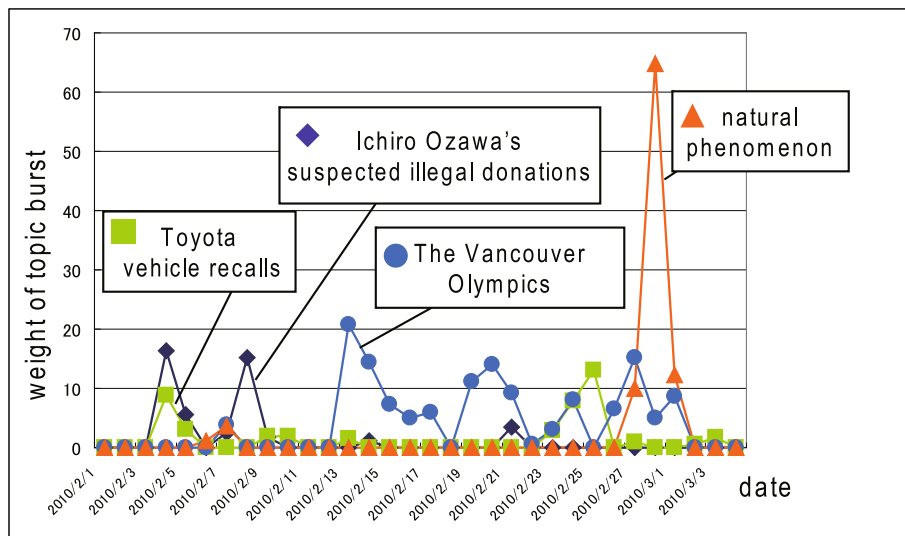


Fig. 6. Changes in the Weight of Topic Bursts

judging the preference among bursty topics through the period of observing those bursty topics. For example, during the period from February 5th to 9th, the two bursty topics “Ichiro Ozawa’s suspected illegal donations” and “Toyota vehicle recalls” are dominant, while from the dates around February 13th when the Vancouver Olympic games started, the topic “the Vancouver Olympics” is clearly dominant. Then, around the dates March 1st, the weight of the topic “natural phenomenon” suddenly rose immediately after the 2010 Chile earthquake occurred on February 27th.

6 Related Works

Compared with related works, the proposed method has its own significance in that it applies the Kleinberg’s burst modeling to statistical time series topic models such as DTM [3]. This paper shows that the Kleinberg’s burst modeling can be easily applied to statistically estimated time series topic models in a quite straightforward fashion.

[4] also employs the Kleinberg’s modeling of keyword burst and applies it to the time series scientific publications. Unlike our approach, however, [4] represent topics in terms of co-occurrence matrix of frequent and bursty keywords. [5] also studied how to rank LDA topics in terms of their significance, although [5] did not study time series document streams nor the issue of bursty topics.

[6] studied how to detect correlated bursty topic patterns across multiple text streams such as multilingual news streams, where their method concentrated on detecting correlated bursty topic patterns based on the similarity of temporal distribution of tokens.

7 Conclusion

This paper focused on two types of modeling of information flow in news stream, namely, burst analysis and topic modeling. This paper especially proposed how to integrate the two types of modeling. We proposed how to apply Kleinberg’s modeling of bursts to topics estimated by a topic model such as DTM. In the evaluation, we showed that the precision of detecting bursty topics is over 83%, which is reasonably high.

Future plans include improving the proposed framework through a larger scale evaluation from various perspectives. Evaluation of recall should be introduced within the overall evaluation procedure. Parameters of topic models such as the number of topics should be examined through further evaluation. Other topic models such as hierarchical ones should be also examined. More theoretical formalization where topic estimation and bursty topic detection are integrated within a single model is also along the direction of future plans. Another issue is how to incorporate online features in the process of detecting bursty topics, where bursty topics should be detected exactly on their early dates when their bursts start without any time series news stream data of the dates after the bursts started. Toward this direction, online topic models such as on-line LDA [7] should be examined.

References

1. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proc. 8th SIGKDD, pp. 91–101 (2002)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022 (2003)
3. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proc. 23rd ICML, pp. 113–120 (2006)
4. Mane, K., Borner, K.: Mapping topics and topic bursts in PNAS. *Proc. PNAS*. 101(suppl. 1), 5287–5290 (2004)
5. AlSumait, L., Barbará, D., Gentle, J., Domeniconi, C.: Topic Significance Ranking of LDA Generative Models. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) *ECML PKDD 2009, Part I. LNCS*, vol. 5781, pp. 67–82. Springer, Heidelberg (2009)
6. Wang, X., Zhai, C., Hu, R.S.: Mining correlated bursty topic patterns from coordinated text streams. In: Proc. 13th SIGKDD, pp. 784–793 (2007)
7. AlSumait, L., Bardara, D., Domeniconi, C.: On-Line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In: Proc. 8th ICDM, pp. 3–12 (2008)

UDRST: A Novel System for Unlabeled Discourse Parsing in the RST Framework

Ngo Xuan Bach, Nguyen Le Minh, and Akira Shimazu

School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{bachnx,nguyenml,shimazu}@jaist.ac.jp

Abstract. This paper presents UDRST¹, an unlabeled discourse parsing system in the RST framework. UDRST consists of a segmentation model and a parsing model. The segmentation model exploits subtree features to rerank N-best outputs of a base segmenter, which uses syntactic and lexical features in a CRF framework. In the parsing model, we present two algorithms for building a discourse tree from a segmented text: an incremental algorithm and a dual decomposition algorithm. Our system achieves 77.3% in the unlabeled score on the standard test set of the RST Discourse Treebank corpus, which improves 5.0% compared to HILDA [6], a state-of-the-art discourse parsing system.

Keywords: Discourse Parsing, Dual Decomposition, Rhetorical Structure Theory, RST, UDRST.

1 Introduction

Discourse parsing is the task of extracting high-level, rhetorical structure in texts, which has been shown to play an important role in many natural language applications, including text summarization [12,14], information presentation [1], and dialogue generation [5]. In the last twenty years, several studies on discourse parsing have been conducted in the framework of Rhetorical Structure Theory (RST) [13], one of the most widely used theories of text structure.

The RST framework consists of two steps: *discourse segmentation* and *discourse tree building*. In the discourse segmentation step, an input text is divided into several elementary discourse units (EDUs). Each EDU may be a simple sentence or a clause in a complex sentence. In the tree building step, consecutive EDUs are put in relation with each other to create a discourse tree. An example of a discourse tree is shown in Figure 1.

Previous studies on discourse parsing reveal two remarkable points. The first point is that the sets of rhetorical relations in different works are inconsistent. For instance, Marcu [14] and Thanh et al. [20] use 15 relations and 14 relations respectively, while Sagae [17] and Hernault et al. [6] use 18 relations. We also

¹ UDRST stands for **U**nla**B**eled **D**iscourse parser in the **RST** framework.

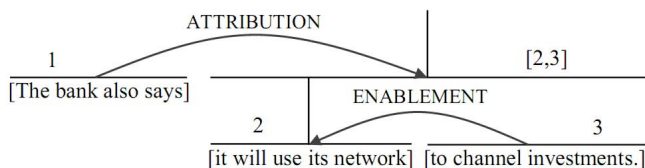


Fig. 1. A discourse tree [18]

note that in RST Discourse Treebank (RST-DT) [2], 78 rhetorical relations are used. The second point is that the performance of a state-of-the-art discourse parsing system is too low when evaluating in the labeled score (both structure and relations). HILDA [6], a state-of-the-art discourse parsing system, achieves only 47.3% in the labeled score on RST-DT. Studies on applications of RST show that in many text analysis applications, only a few relations are enough [14,21]. Furthermore, from a machine learning perspective, working with a small set of relations can improve the performance of a discourse parsing system.

The purpose of this work is to build an unlabeled discourse parsing system, which produces a discourse structure tree without relation labels. The number of relations, types of relations, and the process of labeling relations for the discourse structure tree will be done in text analysis applications, which use this discourse parser. Our discourse parsing system, UDRST, consists of a reranking-based segmentation model and a parsing model using dual decomposition. Experimental results on RST-DT show that UDRST outperforms HILDA [6].

The rest of this paper is organized as follows. Section 2 describes related work on discourse parsing. Section 3 presents background of this study: basic algorithms on discourse parsing (Sub-section 3.1) and the dual decomposition method (Sub-section 3.2). Section 4 describes our discourse parsing system. Experimental results on RST-DT are described in Section 5. Finally, Section 6 gives conclusions.

2 Related Work

Several methods have been proposed to deal with the discourse parsing task. In this section, we present the most related studies, which describe a discourse parsing system in the RST framework.

Soricut and Marcu [18] present a sentence level discourse parser. Two probabilistic models are built to segment and to parse texts. Both models exploit syntactic and lexical information. For discourse segmentation, authors report an F-score of 84.7%. For building sentence level discourse trees, they achieve 70.5% in the unlabeled score, and 49.0% and 45.6% in the labeled score when using 18 labels and 110 labels respectively. However, this discourse parser only processes individual sentences.

Sagae [17] proposes a shift-reduced discourse parser. The parser includes a discourse segmenter based on a binary classifier trained on lexico-syntactic features and a parsing model which employs transition algorithms for dependency and

constituent trees. Compared to Soricut and Marcu [18], the proposed parser is able to create text level discourse trees. The author reports an F-score of 86.7% for discourse segmentation and 44.5% in the labeled score for building text level discourse trees when using 18 labels.

Hernault et al. [6] describe HILDA, a discourse parser using Support Vector Machine (SVM) classification. The parser exploits following kinds of features: textual organization, lexical features, ‘dominance sets’ [18], and structural features. They use 18 relations like relations in the work of Sagae [17]. HILDA is considered as the first fully implemented text level discourse parser with state-of-the-art performance. It achieves 72.3% in the unlabeled score and 47.3% in the labeled score on RST-DT.

3 Background

3.1 Discourse Parsing in the RST Framework

Discourse parsing in the RST framework consists of two steps, *discourse segmentation* and *discourse tree building*. The goal of the discourse segmentation task is to divide an input text into several EDUs. This task is usually considered as a binary classification problem, which assigns a *boundary* label or a *no-boundary* label to each word in the input text [17,18,19]. Another setting for this task is modeling it as a sequence labeling problem [7].

There have been three approaches building a discourse tree given a segmented text. Here we review two approaches[8], which are related to our methods presented in Section 4. The first approach uses a greedy strategy [6]. The method gradually combines two consecutive spans (EDUs or subtrees of EDUs), which are most probably connected by a rhetorical relation, until all EDUs are merged into a single discourse tree.

The tree construction algorithm is presented as Algorithm 1, where l_i denotes the i^{th} element of list L. The algorithm needs a score function (used in lines 4,9, and 10), which evaluates how likely two consecutive spans should be connected. To calculate this score, we first learn a binary classifier *StructClassifier* that takes two consecutive spans as the input and returns +1 in the case two spans should be connected and -1 otherwise. Then we define the score function:

$$StructScore(l_i, l_{i+1}) = Prob(StructClassifier(l_i, l_{i+1}) = +1).$$

Note that if we want to build a labeled tree, in addition to *StructClassifier*, we need a multi-class classifier *LabelClassifier* that also takes two consecutive spans as the input and returns the most probable relation label holding between two spans. In Algorithm 1, we use this classifier to find the relation label before creating a new subtree (line 8 in the algorithm).

The second approach uses a dynamic programming technique [18]. We maintain a two-dimension array $t[i][j]$ storing the most probable structure tree covering from the i^{th} EDU to the j^{th} EDU, and an array $Score[i][j]$ storing the

² The third approach is transition-based discourse parsing [17].

Algorithm 1 A greedy algorithm for discourse tree building [6].

```

1: Input: List of EDUs,  $E = (e_1, e_2, \dots, e_n)$ 
2: Initialize:  $L \leftarrow E$ 
3: for  $(l_i, l_{i+1})$  in  $L$  do
4:    $Scores[i] \leftarrow StructScore(l_i, l_{i+1})$  [Calculate score]
5: end for
6: while  $|L| > 1$  do
7:    $i \leftarrow argmax(Scores)$ 
8:    $NewSubTree \leftarrow CreatTree(l_i, l_{i+1})$  [Create a new subtree]
9:    $Scores[i-1] \leftarrow StructScore(l_{i-1}, NewSubTree)$  [Calculate score]
10:   $Scores[i+1] \leftarrow StructScore(NewSubTree, l_{i+2})$  [Calculate score]
11:   $delete(Scores[i])$  [Update Scores]
12:   $L \leftarrow [l_1, \dots, l_{i-1}, NewSubTree, l_{i+2}, \dots]$  [Update L]
13: end while
14:  $FinalTree \leftarrow l_1$ 
15: Output:  $FinalTree$ .

```

score of $t[i][j]$. The final structure tree is $t[1][n]$, where n is the number of EDUs. $Score[i][j]$ and structure tree $t[i][j]$ can be calculated as follows:

$$Score[i][j] = \max_{i \leq k < j} (Score[i][k] + Score[k+1][j] + StructScore(t[i][k], t[k+1][j]))$$

and

$$t[i][j] \leftarrow CreateTree(t[i][k^*], t[k^*+1][j]),$$

where k^* is the index that maximizes the score function.

Although the first method gives an approximate solution, it is more suitable in practice because it runs much faster than the second method ($O(n)$ compared to $O(n^3)$, where n is the number of EDUs). In fact, the second method is only used in sentence level discourse parsing [18], where the number of EDUs is small.

3.2 Dual Decomposition

Dual decomposition is a method to solve complex optimization problems that can be decomposed into two or more sub-problems, together with linear constraints that enforce the agreement on solutions of the sub-problems [16]. The sub-problems are chosen such that they can be solved efficiently. The constraints are incorporated using Lagrange multipliers, and an iterative algorithm is used to minimize the resulting dual.

We consider the following optimization problem:

$$argmax_{y \in Y, z \in Z} (f(y) + g(z))$$

subject to $y(i) = z(i)$, for all $i \in \{1 \dots n\}$.

Each constraint $y(i) = z(i)$ describes an agreement on the solutions of two sub-problems. We introduce Lagrange multipliers $u(i)$, $i \in \{1 \dots n\}$, and assume that for any value $u(i) \in R$, we can efficiently solve:

$$\operatorname{argmax}_{y \in Y} (f(y) + \sum_{i=1}^n u(i)y(i)), \text{ and}$$

$$\operatorname{argmax}_{z \in Z} (g(z) - \sum_{i=1}^n u(i)z(i)).$$

The dual decomposition algorithm can be expressed as Algorithm 2, where δ_k is the step size at the k^{th} iteration.

Algorithm 2 The dual decomposition algorithm [16].

```

1: Initialize:  $u^{(0)}(i) = 0$ , for all  $i \in \{1 \dots n\}$ 
2: for  $k = 1$  to  $K$  do
3:    $y^{(k)} \leftarrow \operatorname{argmax}_{y \in Y} (f(y) + \sum_{i=1}^n u^{(k-1)}(i)y(i))$  [Sub-problem 1]
4:    $z^{(k)} \leftarrow \operatorname{argmax}_{z \in Z} (g(z) - \sum_{i=1}^n u^{(k-1)}(i)z(i))$  [Sub-problem 2]
5:   if  $y^{(k)}(i) = z^{(k)}(i)$  for all  $i \in \{1 \dots n\}$  then
6:     return  $(y^{(k)}, z^{(k)})$ 
7:   else
8:      $u^{(k)}(i) \leftarrow u^{(k-1)}(i) - \delta_k (y^{(k)}(i) - z^{(k)}(i))$ 
9:   end if
10: end for
11: return  $(y^{(K)}, z^{(K)})$ 

```

Dual decomposition has been applied successfully to several NLP tasks such as parsing [15], dependency parsing [9], and coordination disambiguation [4].

4 UDRST: An Unlabeled Discourse Parsing System

This section describes our discourse parsing system, UDRST. We first present our discourse segmenter using a reranking method in Sub-section 4.1. We then describe two algorithms for building discourse trees in Sub-section 4.2.

4.1 A Model for Discourse Segmentation

Discriminative Reranking. In the discriminative reranking method [3], first, a set of candidates is generated using a base model (GEN). GEN can be any model for the task. Then, candidates are reranked using a linear score function:

$$\operatorname{score}(y) = \Phi(y) \cdot W$$

where y is a candidate, $\Phi(y)$ is the feature vector of candidate y , and W is a parameter vector. The final output is the candidate with the highest score:

$$F(x) = \operatorname{argmax}_{y \in \operatorname{GEN}(x)} \operatorname{score}(y) = \operatorname{argmax}_{y \in \operatorname{GEN}(x)} \Phi(y) \cdot W.$$

To learn the parameter W we use the average perceptron algorithm [3].

Base Model. Similar to the work of Hernault et al. [7], our base model uses Conditional Random Fields (CRFs)³ to learn a sequence labeling model. Each label is either *beginning* of EDU (B) or *continuation* of EDU (C).

We use the following lexical and syntactic information as features: words, POS tags, nodes in parse trees and their lexical heads and their POS heads. When extracting features for word w , let r be the word on the right-hand side of w and N_p be the deepest node that belongs to both paths from the root to w and r . N_w and N_r are child nodes of N_p that belong to two paths, respectively. Figure 2 shows two partial lexicalized syntactic parse trees. In the first tree, if $w = \textit{says}$ then $r = \textit{it}$, $N_p = VP(\textit{says})$, $N_w = VBZ(\textit{says})$, and $N_r = SBAR(\textit{will})$. We also consider the parent and the right-sibling of N_p if any. The final feature set for w consists of not only features extracted from w but also features extracted from two words on the left-hand side and two words on the right-hand side of w .

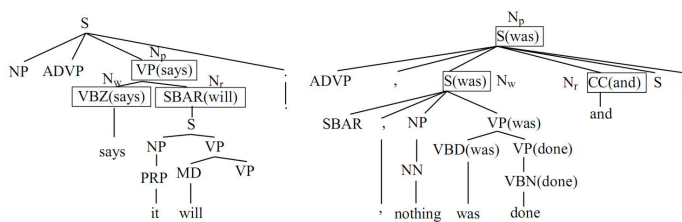


Fig. 2. Partial lexicalized syntactic parse trees

Our feature extraction method is different from the method in previous work [7][8]. They define N_w as the highest ancestor of w that has lexical head w and has a right-sibling. Then N_p and N_r are defined as the parent and right-sibling of N_w . In the first example, our method gives the same results as the previous one. In the second example, however, there is no node with lexical head “done” and having a right-sibling. The previous method cannot extract N_w , N_p , and N_r in such cases. We also use some new features such as the head node and the right-sibling node of N_p .

Subtree Features for Reranking. We need to decide which kinds of subtrees are useful to represent a candidate, a way to segment the input sentence into EDUs. In our work, we consider two kinds of subtrees: *bound trees* and *splitting trees*.

The *bound tree* of an EDU, which spans from word u to word w , is a subtree which satisfies two conditions: 1) its root is the deepest node in the parse tree which belongs to both paths from the root of the parse tree to u and w ; and 2) it only contains nodes in two those paths.

The *splitting tree* between two consecutive EDUs, from word u to word w and from word r to word v , is a subtree which is similar to a bound tree, but contains

³ We use the implementation of Kudo [10].

two paths from the root of the parse tree to w and r . Bound trees will cover the whole EDUs, while splitting trees will concentrate on the boundaries of EDUs.

From a bound tree (similar to a splitting tree), we extract three kinds of subtrees: subtrees on the left path (*left tree*), subtrees on the right path (*right tree*), and subtrees consisting of a subtree on the left path and a subtree on the right path (*full tree*). In the third case, if both subtrees on the left and right paths do not contain the root node, we add a pseudo root node. Figure 3 shows the bound tree of EDU “nothing was done” of the second example in Figure 2, and some examples of extracted subtrees.

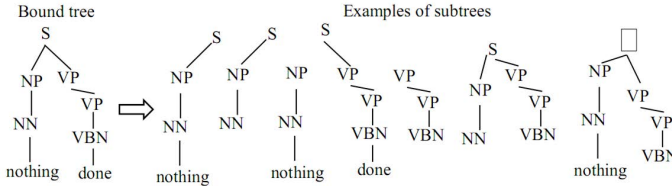


Fig. 3. Subtree features

The feature set of a candidate is the set of all subtrees extracted from bound trees of all EDUs and splitting trees between two consecutive EDUs.

4.2 Two Algorithms for Building Discourse Trees

In the tree building step, the goal is to build a discourse tree given a text which has been segmented into EDUs. Usually, the text consists of several paragraphs, and each paragraph consists of some sentences. We note that EDUs within one sentence tend to be connected to make a subtree before connecting to EDUs in other sentences. The same thing also takes place at the paragraph level. EDUs within one paragraph tend to be connected to make a subtree before connecting to EDUs in other paragraphs. It is because of the coherence of a well-written text. A sentence expresses a statement, question, exclamation, request, command, or suggestion, and sentences in a paragraph should focus on a topic. Paragraphs help separate ideas and indicate the change of topics.

From this property, we propose an incremental algorithm (our first algorithm) for building a discourse tree. The algorithm is presented as Algorithm 3. When we build a subtree for a sentence, the input consists of EDUs in that sentence. When we build a subtree for a paragraph, the input consists of several subtrees, and each subtree corresponds to a sentence. When we build the final discourse tree for whole text, the input consists of several subtrees, and each subtree corresponds to a paragraph. In all three cases, the number of spans (EDUs or subtrees) in the input is very small in comparison with total EDUs of whole text. So we can use a dynamic programming technique like the method presented in Section 3.1 [18] to solve it.

Compared to the greedy algorithm (Algorithm 1) presented in Section 3.1 [6], this algorithm has two advantages. It supports the coherence property we

Algorithm 3 An incremental algorithm for building discourse trees.

```

1: Input: a text  $T$ 
2: for each sentence  $s$  in  $T$  do
3:   Create a subtree for  $s$ 
4: end for
5: for each paragraph  $p$  in  $T$  do
6:   Create a subtree for  $p$  based on subtrees of sentences
7: end for
8: Create a discourse tree for  $T$  based on subtrees of paragraphs
9: Output: Discourse tree

```

described before. The algorithm employs a dynamic programming technique, so it can find an extract solution in each step. However, the algorithm gives a hard constraint on the order in which EDUs are connected. So it makes the search process biased. Our solution is creating a parsing model by integrating two above algorithms. To achieve this, we employ dual decomposition.

Recall that the parsing problem is to find:

$$\operatorname{argmax}_{y \in Y} h(y)$$

where Y is the space of all possible discourse trees, and $h(y)$ is a score function defined on Y . In our method, the score function consists of two factors $h(y) = f(y) + g(y)$, where $f(y)$ is the score returned by our base model1 using the incremental algorithm (Algorithm 3), and $g(y)$ is the score returned by our base model2 using the greedy algorithm (Algorithm 1).

For each discourse tree y , we define variables $y(i, j)$ as follows:

$$y(i, j) = \begin{cases} 1 & \text{if exists a subtree that covers from the } i^{\text{th}} \text{ EDU to the } j^{\text{th}} \text{ EDU} \\ 0 & \text{otherwise.} \end{cases}$$

The problem now becomes:

$$\operatorname{argmax}_{y \in Y, z \in Z} (f(y) + g(z))$$

subjects to: $y(i, j) = z(i, j)$ for all $1 \leq i < j \leq n$, where n is the number of EDUs and $Z = Y$. We solve this problem by using dual decomposition. The proposed algorithm (our second algorithm) is presented as Algorithm 4, where $u(i, j)$ are Lagrange multipliers.

Note that when solving two sub-problems (lines 3 and 4 in Algorithm 4) using two base algorithms (Algorithms 1 and 3), the only thing we need to modify is the score functions. They will become:

$Score[i][j] = \max_{i \leq k < j} (Score[i][k] + Score[k+1][j] + StructScore(t[i][k], t[k+1][j])) + \mathbf{u}(\mathbf{i}, \mathbf{j})$, in base model1, and

$Score[i][j] = \max_{i \leq k < j} (Score[i][k] + Score[k+1][j] + StructScore(t[i][k], t[k+1][j])) - \mathbf{u}(\mathbf{i}, \mathbf{j})$, in base model2.

To learn the binary classifier *StructClassifier*, which is used to compute *StructScore*, like Hernault et al. [6], we use lexical and syntactic features including textual organization features, lexical features, ‘dominance sets’ [18], and

Algorithm 4 A dual decomposition algorithm for building discourse trees.

```

1: Initialize:  $u^{(0)}(i, j) = 0$ , for all  $1 \leq i < j \leq n$ .
2: for  $k = 1$  to  $K$  do
3:    $y^{(k)} \leftarrow \operatorname{argmax}_{y \in Y} (f(y) + \sum_{1 \leq i < j \leq n} u^{(k-1)}(i, j) y(i, j))$  [Base Model1]
4:    $z^{(k)} \leftarrow \operatorname{argmax}_{z \in Z} (g(z) - \sum_{1 \leq i < j \leq n} u^{(k-1)}(i, j) z(i, j))$  [Base Model2]
5:   if  $y^{(k)}(i, j) = z^{(k)}(i, j)$  for all  $1 \leq i < j \leq n$  then
6:     return  $y^{(k)}$ 
7:   else
8:      $u^{(k)}(i, j) \leftarrow u^{(k-1)}(i, j) - \delta_k (y^{(k)}(i, j) - z^{(k)}(i, j))$ 
9:   end if
10: end for
11: return  $y^{(K)}$ 

```

structural features. We also employ Support Vector Machines⁴ as the learning method.

5 Experiments

5.1 Corpus and Evaluation Method

We tested our system on the RST Discourse Treebank corpus. This corpus consists of 385 articles from the Penn Treebank, which are divided into a Training set and a Test set. The Training set consists of 347 articles (6132 sentences), and the Test set consists of 38 articles (991 sentences).

For the discourse segmentation task, there are two evaluation methods that have been used in previous work. The first method measures only *beginning* labels (B labels) [18,19]. The second method [6,7] measures both *beginning* and *continuation* labels (B and C labels)⁵. This method first calculates scores on B labels and scores on C labels, and then produces the average of them. Due to the number of C labels being much higher than the number of B labels, the second evaluation method yields much higher results. In our work, we measure the performance of the proposed model using both methods. For the discourse parsing task, we measure the performance of the proposed system using the unlabeled score, which is the same as the unlabeled score described in previous work [6,14,18].

5.2 Experimental Results on Discourse Segmentation

We learned the base model on the Training set and tested on the Test set to get N-best outputs to rerank. To learn parameters of the reranking model, we conducted 5-fold cross-validation tests on the Training set. In all experiments,

⁴ In our experiments, we used Libsvm: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

⁵ Neither evaluation method counts sentence boundaries.

we set N to 20. To choose the number of iterations, we used a development set, which is about 20 percent of the Training set.

Table 1 shows experimental results of discourse segmenters on RST-DT, in which SPADE is the work of Soricut and Marcu [18], NNDS is a segmenter that uses neural networks [19], HILDA-Seg is the work of Hernault et al. [6], CRFSeg is a CRF-based segmenter [7], Base is our base model, and UDRST-Seg is our model using reranking.

When evaluating on B labels, our base model got 92.5% and 90.7% in two settings using gold parse trees and Stanford parse trees [8], which improves 1.3% and 1.7% compared to the state-of-the-art segmenter (CRFSeg). It demonstrates the effectiveness of our feature extraction method in the base model. As expected, UDRST-Seg got higher results compared to the base model in both settings. UDRST-Seg achieved 93.7% and 91.0% in two settings, which improves 2.5% and 2.0% compared to CRFSeg. Also note that, when using Stanford parse trees, UDRST-Seg got competitive results with CRFSeg when using gold parse trees (91.0% compared to 91.2%). When evaluating on B and C labels, UDRST-Seg also outperforms CRFSeg in both settings, using gold parse trees and Stanford parse trees (96.6% and 95.1% compared to 95.3% and 94.1% of CRFSeg).

Table 1. Performance of discourse segmenters

		Evaluating on B labels			Evaluating on B and C labels		
Model	Trees	Precision(%)	Recall(%)	F_1 (%)	Precision(%)	Recall(%)	F_1 (%)
SPADE	Penn	84.1	85.4	84.7	-	-	-
NNDS	Penn	85.5	86.6	86.0	-	-	-
HILDA-Seg	Penn	-	-	-	95.5	94.5	95.0
CRFSeg	Penn	92.7	89.7	91.2	96.0	94.6	95.3
Base	Penn	92.5	92.5	92.5	96.0	96.0	96.0
UDRST-Seg	Penn	93.1	94.2	93.7	96.3	96.9	96.6
HILDA-Seg	Stanford	-	-	-	94.5	93.1	93.8
CRFSeg	Stanford	91.0	87.2	89.0	95.0	93.2	94.1
Base	Stanford	91.4	90.1	90.7	95.3	94.7	95.0
UDRST-Seg	Stanford	91.5	90.4	91.0	95.4	94.9	95.1

5.3 Experimental Results on Discourse Parsing

We tested our system on the Test set of RST-DT in two settings. In the first setting, we used gold segmentation and Penn Treebank parse trees. The purpose of this setting is to test the performance of the proposed parsing model. In the second setting, we used segmentation produced by our discourse segmenter and Stanford parse trees. The purpose of this setting is to test the performance of the full system. In all experiments, the step size δ_k was chosen as the guidance in Rush and Collins [16], and the number of iterations K was set to 10.

Table 2 shows experimental results of the tree building step in the unlabeled score. When using the incremental algorithm, UDRST achieved 84.3% in the F_1 score, which improves 1.3% compared to HILDA. As expected, UDRST with dual

Table 2. Experimental results of the tree building step (gold segmentation and gold parse trees)

System	Algorithm	Precision(%)	Recall(%)	F_1 (%)	Improvement(%)
HILDA	Greedy	83.0	83.0	83.0	-
UDRST	Incremental	84.3	84.3	84.3	1.3
	Dual	84.6	84.6	84.6	1.6

decomposition algorithm got the better result than UDRST with the incremental algorithm (84.6% compared to 84.3%).

Table 3 shows the performance of the full system in the unlabeled score. UDRST outperforms HILDA in both algorithms, the incremental algorithm and the dual decomposition algorithm. It achieved 77.0% and 77.3% in two algorithms, which improve 4.7% and 5.0% compared to HILDA.

Table 3. Performance of the full system (our segmentation model and Stanford parse trees)

System	Algorithm	Precision(%)	Recall(%)	F_1 (%)	Improvement(%)
HILDA	Greedy	73.0	71.7	72.3	-
UDRST	Incremental	77.2	76.7	77.0	4.7
	Dual	77.5	77.0	77.3	5.0

We do not compare our system to systems described in [17,18]. Sagae [17] does not report the performance of his system in the unlabeled score. Soricut and Marcu [18] evaluate their system only on sentence level discourse parsing. They achieve 70.5% in the unlabeled score.

6 Conclusion

We presented a novel system for the discourse parsing task in the RST framework. Our discourse parser, UDRST, consists of a discourse segmenter using a reranking method with subtree features and a parsing model which employs dual decomposition. The basic idea of the parsing model is that EDUs in a sentence (a paragraph) tend to be connected to form a subtree before connecting to EDUs in other sentences (paragraphs). This idea is put into our model by integrating an incremental model and a greedy model using dual decomposition. Experiments on the RST Discourse Treebank corpus show that UDRST outperforms HILDA [6], a state-of-the-art discourse parsing system in the unlabeled score.

References

1. Bateman, J., Klein, J., Kamps, T., Reichenberger, K.: Towards Constructive Text, Diagram, and Layout Generation for Information Presentation. *Computational Linguistics* 27(3), 409–449 (2001)

2. Carlson, L., Marcu, D., Okurowski, M.E.: RST Discourse Treebank. Linguistic Data Consortium, LDC (2002)
3. Collins, M., Koo, T.: Discriminative Reranking for Natural Language Parsing. *Computational Linguistics* 31(1), 25–70 (2005)
4. Hanamoto, A., Matsuzaki, T., Tsujii, J.: Coordination Structure Analysis using Dual Decomposition. In: *Proceedings of EACL*, pp. 430–438 (2012)
5. Hernault, H., Piwek, P., Prendinger, H., Ishizuka, M.: Generating Dialogues for Virtual Agents Using Nested Textual Coherence Relations. In: Prendinger, H., Lester, J.C., Ishizuka, M. (eds.) *IVA 2008. LNCS (LNAI)*, vol. 5208, pp. 139–145. Springer, Heidelberg (2008)
6. Hernault, H., Prendinger, H.A., Du Verle, D., Ishizuka, M.: HILDA: A Discourse Parser Using Support Vector Machine Classification. *Dialogue and Discourse* 1(3), 1–33 (2010)
7. Hernault, H., Bollegala, D., Ishizuka, M.: A Sequential Model for Discourse Segmentation. In: Gelbukh, A. (ed.) *CICLing 2010. LNCS*, vol. 6008, pp. 315–326. Springer, Heidelberg (2010)
8. Klein, D., Manning, C.: Accurate Unlexicalized Parsing. In: *Proceedings of ACL*, pp. 423–430 (2003)
9. Koo, T., Rush, A.M., Collins, M., Jaakkola, T., Sontag, D.: Dual Decomposition for Parsing with Non-Projective Head Automata. In: *Proceedings of EMNLP*, pp. 1288–1298 (2010)
10. Kudo, T.: CRF++: Yet Another CRF toolkit, <http://crfpp.sourceforge.net/>
11. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proceedings of ICML*, pp. 282–289 (2001)
12. Louis, A., Joshi, A., Nenkova, A.: Discourse indicators for content selection in summarization. In: *Proceedings of SIGDIAL*, pp. 147–156 (2010)
13. Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory. *Toward a Functional Theory of Text Organization. Text* 8, 243–281 (1988)
14. Marcu, D.: *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge (2000)
15. Rush, A.M., Sontag, D., Collins, M., Tommi, J.: On Dual Decomposition and Linear Programming Relaxations for Natural Language Processing. In: *Proceedings of EMNLP*, pp. 1–11 (2010)
16. Rush, A.M., Collins, M.: A Tutorial on Dual Decomposition and Lagrangian Relaxation for Inference in Natural Language Processing. Tutorial at *ACL* (2011)
17. Sagae, K.: Analysis of Discourse Structure with Syntactic Dependencies and Data-Driven Shift-Reduce Parsing. In: *Proceedings of IWPT*, pp. 81–84 (2009)
18. Soricut, R., Marcu, D.: Sentence Level Discourse Parsing using Syntactic and Lexical Information. In: *Proceedings of NAACL*, pp. 149–156 (2003)
19. Subba, R., Di Eugenio, B.: Automatic Discourse Segmentation using Neural Networks. In: *Proceedings of SemDial*, pp. 189–190 (2007)
20. Thanh, H.L., Abeyasinghe, G., Huyck, C.: Generating Discourse Structures for Written Texts. In: *Proceedings of COLING*, pp. 329–335 (2004)
21. Zirn, C., Niepert, M., Stuckenschmidt, H., Strube, M.: Fine-Grained Sentiment Analysis with Structural Features. In: *Proceedings of IJCNLP*, pp. 336–344 (2011)

Long-Term Goal Discovery in the Twitter Posts through the Word-Pair LDA Model

Dandan Zhu¹, Yusuke Fukazawa², Eleftherios Karapetsas¹, and Jun Ota¹

¹ The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, 277-8568 Japan
{zhu, eleftherios, ota}@race.u-tokyo.ac.jp

² Services & Solution Development Dept., NTTDOCOMO, Inc., NTT DOCOMO R&D Center
3-5 Hikarinooka, Yokosuka Kanagawa, 239-8536 Japan
fukazawayuu@nttdocomo.co.jp

Abstract. We used the twitter posts about New Year's resolutions as data source to capture users' long-term goals. New Year's resolutions are the commitments that people set for their personal goals, and generally, people plan to fulfill them for the whole following year. Therefore, we can think of such tweets as data source to explore people's possible long-term goals. The key words in each tweet were extracted for clustering. Considering the form of word-pairs led by verbs is a more intuitive and clearer way to express people's intentions than the one of separate words, we propose a generative model that incorporates word connections into the smoothed LDA to cluster the key words of long-term goals. The experiments demonstrate the proposed model is capable of clustering the word-pairs with better intuitive character, and clearly dividing people's long-term goals.

Keywords: Long-term goals, twitter posts, LDA model, word-pairs.

1 Introduction

Recommender systems have been integrated into our lives since people depend more and more on internet to solve diverse problems. Through predicting users' demands, recommender systems can customize and provide information services. However, most of them just focus on users' temporary queries, and prefer to offer information with high confidence level, naturally ignoring users' deep intentions. Therefore, what they could provide are usually too obvious, which likely means low-value recommendations for users.

Our future goal is to build a recommender system that is capable of offering serendipitous information according to the user's long-term goals, and could serve people in a continuous, enduring and comprehensive way. It can be considered that people's single effort may reflect their long-term goals more or less, and conversely, knowing their long term goals is critical to better understand their behaviors and service them with a broad vision. Imagine the feelings if a search engine could understand your long-term goals: it knows what would be needed for your future development, and could serve you with searching results that would solve your potential problems instead of just current ones.

The crucial part of the proposed recommender system is the long-term goal dataset which would be used for reference to tell the categories that users' goals belong to. This dataset is the concern of this paper. We used twitter posts about New Year's resolutions as long-term goal sources, and clustered the processed data into various topic-classes. The proposed model for clustering, named word-pair LDA model (wpLDA), incorporates the relatedness of words into the LDA, and could assign word-pairs to the according topic-classes.

The task of extracting a sequence of words or terms that co-occur more often than would be expected by chance is called collocation extraction, and it has been used in topic models to enhance the clustering effects of LDA model[1,2,3]. However, most of words found together by these techniques are often the elements of compound nouns or phrasal verbs, meaning each combination present only one part of speech. For example, "electric toothbrush" is a noun, though its compositions are an adjective and a noun. Therefore, such kinds of collocations are still separate words for our purpose.

The most related research is the Must-Link[4], however, the relation it built is the synonymous link, and thus, verbs can be only associated with synonymous verbs which seem to appear with a similarly high probability in the same topics. Therefore, their approach is inappropriate for the purpose of generating the intuitive expression in the form of word-pairs like a verb plus a noun, whose pairs are not with synonymous relationship.

2 Data Retrieve and Processing

In the past few years, social networks like *Twitter* have been experiencing an explosive growth, and many people like posting their thoughts on social networks. And peculiarly, during New Year, which is an intensive phase of release, you can easily find messages about New Year's resolutions on social networks. In General, New Year's resolutions are the commitments that people set for their personal goals, and people plan to fulfill them for the whole following year. Therefore, considering plans for the whole year as long term goals, we treat them as quite suitable data for directly express people's various long-term goals.

In this research, we used the tweets about users' New Year's resolutions to build the long-term goal dataset. We searched for the tweets on the search engines, *Bing*, *Google* and *Yahoo*, under the queries "new year(s)(s) resolution(s) is(are) to". The total collection is 1488 pieces of tweets after eliminating the duplicates.

Then, the goal-related descriptions in the raw data were extracted. By removing the words before "is to" or "are to" (including "is to" and "are to") and the words after the first stop symbols, such as period and exclamation mark, we caught the key words that directly reflect user's new year's resolution for each tweet.

Generally, effective expression of intentions requires a verb and an object. The object is usually denoted as a noun, which could be a person, a place or a thing, while the role of verbs is to illustrate the behaviors toward objects. One of the advantages of this combination is that it makes information transfer more efficient and with less

boundaries. Besides nouns, there are some other words, especially adjectives and adverbs, which may also play an important role in communication, sometimes even more useful than nouns in identifying the message conveyed. For example, if a man’s wish is to “eat healthier food”, then we can more accurately tell his intention through the verb-adjective pair “eat healthier” than the verb-noun pair “eat food”. To prevent missing any possible functional form of pairs, we built the connection between verbs and non-verb words, and it is reasonable that the words in pairs should share the same topics. By using Helmut Schmid’s *TreeTagger* [5], we separated verbs from other words and used these tagged data as the input of the wpLDA model.

3 Word-Pair LDA Model

As we mentioned in Section 2, the words in the verb-non-verb pairs are supposed to be assigned to the same topics, so we proposed the wpLDA model to fulfill this task.

Let $supp(X)$ be the support of set X , we use the smoothed “confidence” to measure the relatedness of words in a word-pair (ω_i, ω_j) :

$$\mu_{i,j} = \frac{supp(\omega_i \cap \omega_j) + \lambda}{supp(\omega_i) + \lambda * M} \tag{1}$$

where λ is the smoothing coefficient, M is the number of documents.

The topic-specific word-pair probability distribution is described as a connection lattice, where the probability that a word-pair (i, j) assigned to the topic k is $\gamma_{k(i,j)} = \rho \cdot \varphi_{k,i} \cdot \mu_{i,j} \cdot \varphi_{k,j}$, in which ρ is the normalization constant.

The generation factor graph of the proposed model is given in Fig. 1. The generative process for each document d in a corpus is similar to the standard LDA but for generating word-pairs instead of separate words: when generate a word ω_1 , we consider all the possible partner for pairing, and assign a latent topic to this word-pair.

The generative process for each document j in the corpus is as follows:

1. Draw $\vartheta_j \sim Dir(\alpha)$, where $j \in \{1, \dots, M\}$, M is the number of documents in the corpus, and $Dir(\alpha)$ is the Dirichlet distribution for parameter α ;
2. Draw $\varphi_k \sim Dir(\beta)$, where $k \in \{1, \dots, K\}$, K is the number of topics, and $Dir(\beta)$ is the Dirichlet distribution for parameter β ;
3. For each of word-pair $pr_{j,t}$, where the word-pair token $t \in \{1, \dots, N_d\}$, and N_d is the number of words in document j :
 - a. Draw a topic $z_{j,t} \sim Multinomial(\vartheta_j)$;
 - b. Draw a word-pair $pr_{j,t} \sim Multinomial(\gamma_{z_{j,t}})$.

By using the Collapsed Gibbs Sampling, the recursion formula is obtained as follows:

$$\propto \left(n_{m,(.)}^{k,-(m,n)} + \alpha_k \right) \cdot \frac{P(z_{(m,n)} = k | z_{-(m,n)}, PR; \alpha, \beta, \mu)}{\sum_{r=1}^V \left(n_{(.),r}^{k,-(m,n)} + \beta_r \right) \left(\sum_{r=1}^V \left(n_{(.),r}^{k,-(m,n)} + \beta_r \right) + 1 \right)} \cdot \mu_{k,(v_1,v_2)}^{n_{(.),v}^{k,-(m,n)}} \tag{2}$$

where the value of parameters, $n_{m,(.)}^{k,-(m,n)}$, $n_{(.),v_1}^{k,-(m,n)}$, $n_{(.),v_2}^{k,-(m,n)}$, $\sum_{r=1}^V n_{(.),r}^{k,-(m,n)}$ and $n_{(.),v'}^{k,-(m,n)}$ can be obtained by simply counting the current topic assignment in the corpus.

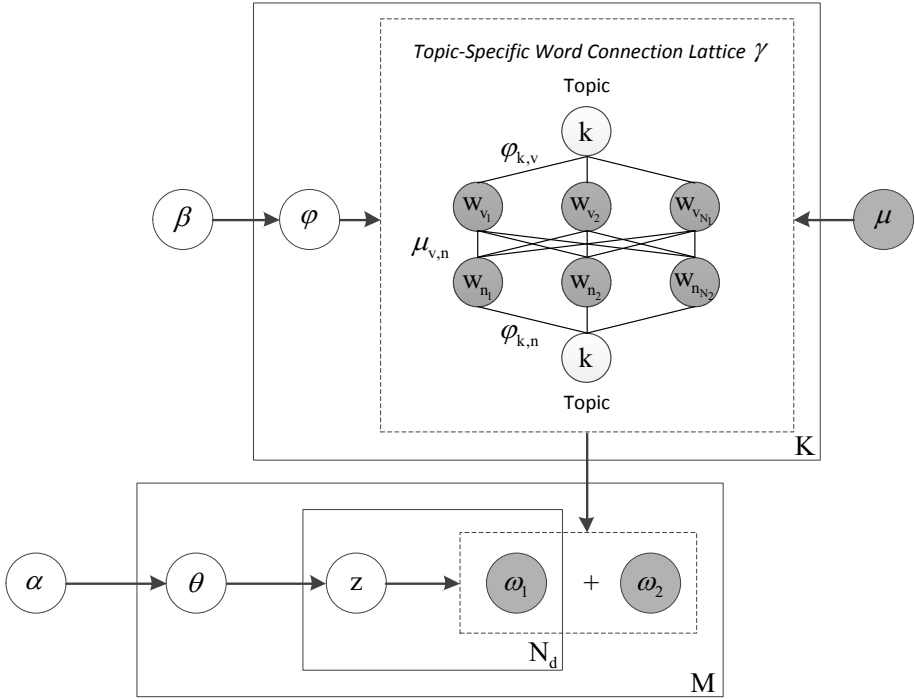


Fig. 1. Plate notation for the generation of word-pairs

Table 1. Definition of Variables in the Model

Variables	Meaning
K	Number of topics
N_d	Number of word-pairs in a document d
α	Parameter of the Dirichlet prior on θ
β	Parameter of the Dirichlet prior on ϕ
θ	Per-document topic probability distribution
ϕ	Per-topic word probability distribution
Z	Topic for a word-pair
γ	Per-topic word-pair probability distribution
$Z_{(m,n)}$	The topic of the n^{th} word-pair (v_1, v_2) in document m
$Z_{-(m,n)}$	All the topics of word-pairs but $Z_{(m,n)}$
PR	All the word-pairs in the corpus
$n_{m,(.)}^{k,-(m,n)}$	Number of words assigned to the k^{th} topic in the m^{th} document
$n_{(.),v_1}^{k,-(m,n)} / n_{(.),v_2}^{k,-(m,n)} / n_{(.),r}^{k,-(m,n)} / n_{(.),v'}^{k,-(m,n)}$	Number of verb v_1 (non-verb v_2 , word r , or word-pair v') assigned to the k^{th} topic

4 Experiments

To evaluate the sensitivity of the proposed model, we varied the number of topics from 5 to 50, and used the perplexity as the evaluation criterion:

$$perplexity = \exp \left\{ -\frac{\sum_{d=1}^M \log p(pr_d)}{\sum_{d=1}^M N_d} \right\} \quad (3)$$

where $p(pr_d)$ is the generative probability of all the word-pairs in the document d .

The best value for the number of topics is about 20, since its perplexity is near to the minimum. And after about 15 iterations, the perplexity enters a stable state, suggesting that the algorithm converges after about 15 iterations' operation.

A baseline version of wpLDA has been built to figure out how the relatedness effects on the clustering of word-pairs. It differs from the wpLDA model in two ways: $\mu_{i,j}$ takes the minimum of (1), which equals to $1/2M$; a constant value $1/V$ is used as the probability of non-verbs' topics instead of (2), which is the minimum probability that a word is assigned to a topic. These settings have blunted the relatedness of words in pairs.

Table. 2 is the comparison of the baseline wpLDA and the wpLDA on perplexity. It shows that the baseline model has higher value of perplexity, meaning the results share lower likelihood. This is because the baseline model equally treats the relatedness between words in pairs, and meanwhile it greatly weakens the relatedness of words, which reduce the likelihood of the classes.

Table 2. Perplexity of the baseline-wpLDA and wpLDA (No. of class=20)

Model	Iterations						
	1	4	7	10	13	16	19
Baseline	8769.	6722.	6489.	6280.	6860.	6681.	5847.
wpLDA	4936.	3945.	3985.	3838.	3989.	3523.	3463.

To verify the effect of word-pair clustering, we used “pair-weight” as the criterion to select best word-pairs for each class. The pair-weight under a topic is defined as follow:

$$pair - weight(\omega_i, \omega_j) = conf(\omega_i, \omega_j) \cdot N_{\omega_i} \cdot N_{\omega_j} \quad (4)$$

where ω_i is a verb, ω_j is a non-verb word, N_{ω} is the number of appearance of ω in this class.

Table 3 lists 5 classes of the word-pairs with the highest value of pair-weight, which produced by the baseline wpLDA and the wpLDA, respectively. Since the baseline model ignores the relatedness of word-pairs, some unexpected words are set as pairs, such as “cut_gym” and “stop_day”, while the results of the proposed model have less such pairs.

In addition, the results show that the average value of pair-weights in the classes of wpLDA is higher than the one of the baseline model by 3.2%, which means the classes of wpLDA are tighter than the ones of the baseline model. As can be seen from Table 3, the classes of wpLDA capture similar tweets under each topic, and they are able to recapitulate users' main ideas of tweets, covering various aspects of people's long-term goals and demonstrating excellent intuitive character.

Table 3. Word-Pair Classes: The Baseline wpLDA and The wpLDA (Snippets)

Baseline wpLDA				
cut_week	lose_diet	stop_best	like_person	come_word
study_week	lose_weight	stop_people	like_people	stop_day
write_week	quit_weight	stop_life	like_phone	help_really
start_week	quit_diet	stop_time	like_best	help_day
cut_days	help_weight	stop_word	talk_person	stop_fun
cut_gym	help_diet	stop_fat	live_person	help_fun
cut_way	lose_smoking	stop_fucking	love_person	use_really
cut_just	watch_diet	beat_best	talk_best	stop_word
wpLDA				
love_things	exercise_diet	quit_smoking	spend_time	make_business
like_things	start_diet	quit_finally	spend_friends	make_best
love_person	eat_diet	eat_good	spend_god	make_fat
love_getting	exercise_healthier	stop_good	gain_time	make_getting
know_things	exercise_life	stop_smoking	buy_time	make_organized
live_things	live_healthier	stop_finally	spend_actually	fit_business
read_things	live_life	watch_smoking	spend_internet	save_organized
work_things	eat_healthier	watch_good	spend_food	fit_best

5 Conclusions

We have explored people’s long-term goals by focusing on the naturally occurring information on social networks. The key words of people’s long-term goals were extracted from the tweets about New Year’s resolutions, and then we used the wpLDA model to cluster word-pairs into various topics to create a long-term goal dataset. The experiment results demonstrate that the wpLDA is able to clearly and intuitively divide people’s long-term goals. This work is a preliminary one for further study of people’s long term goals.

It is important to note that the association rules we used to build words’ relatedness is replaceable, and the word-pair could be any combination of words, which means the proposed model is a general model and could be used for multiple purposes.

References

1. Erosheva, E., Fienberg, S., Lafferty, J.: Mixed-membership models of scientific publications. Proc. of the National Academy of Sciences, 5220–5227 (2004)
2. Hoffman, T.: Probabilistic Latent Semantic Analysis. In: Proc. of Uncertainty in Artificial Intelligence, UAI (1999)
3. Nallapati, R., Ahmed, A., Xing, E.P., Cohen, W.W.: Joint Latent Topic Models for Text and Citations. In: Proc. of KDD, pp. 24–27 (2008)
4. Andrzejewski, D., Zhu, X., Craven, M.: Incorporating Domain Knowledge into Topic Modeling via Dirichlet Forest Priors. In: Proc. of ICML (2009)
5. Schmid, H.: Improvements in Part-of-Speech Tagging with an Application to German. In: Proc. of the ACL SIGDAT-Workshop, pp. 47–50 (1995)

Finding Social Relationships by Extracting Polite Language in Micro-blog Exchanges

Norinobu Hatamoto, Yoshiaki Kurosawa, Shogo Hamada, Kazuya Mera,
and Toshiyuki Takezawa

Graduate School of Information Sciences, Hiroshima City University, Hiroshima, Japan
{hatamoto, kurosawa, hamada, mera,
takezawa}@ls.info.hiroshima-cu.ac.jp

Abstract. The aim of this study was to describe user relationships based on honorific expressions in messages posted to a micro-blogging service and to classify the users into appropriate groups. In particular, we focused on attitudinal expressions that indicate the speaker's attitude. We compiled posts on micro-blogging platform Twitter and performed an experiment to classify the data based on honorifics. In the results, compared with indegree centrality values, the obtained social graph was superior to one acquired from a baseline, i.e., by the condition of the follower-followed relationship.

Keywords: micro-blogging service, social relationships, soft clustering, polite language, attitudinal expressions.

1 Introduction

The recent development of the Internet and appearance of new communication tools on the Internet has dramatically changed our daily lives. Although there are many such tools in the world, we focus on Twitter¹, a social networking service and one of the most popular micro-blogging services.

Twitter enables people to communicate with each other through the action of 'tweeting', which means posting a short message, or 'tweet', to a specific person or anyone and everyone. Through this action, users can be tied to other users, regardless of the real-world connection. This tie leads participants to belong to different communities where they have various relationships. For example, if we focus on a university, we can easily see the relationships 'classmates', 'colleagues', 'professor-student', 'senior-junior', and so on. These relationships include vertical ones in society as well as horizontal ones, such as friendship. These relationships can be constructed easily in Twitter.

As various relationships exist in Twitter, the question then becomes whether we can extract such relationships. We aim to do so, in particular attempting to address vertical relationships in society such as 'professor-student' and 'senior-junior'.

To extract these relationships, we focus on attitudinal expressions that indicate the speakers' attitude. In Japanese, expressions such as '*desu*' and '*masu*' appear in

¹ <http://twitter.com>

conversation. These expressions implicitly indicate the vertical and hierarchical status of the speaker. They are used when speaking to an elderly person or a person in a high position; not doing so is very impolite. That is, the appearance of these expressions is important information when extracting relationships among users. The direction in which attitudinal expressions are uttered is also important for differentiating the status of the users.

Receiving many attitudinal expressions would indicate that the receiver is in a higher position. He or she may be a powerful person in the community to which he or she belongs. For this reason, we focus on attitudinal expression in tweets and attempt to extract influential people. This extraction is very important because it helps us to propose certain tweets on certain topics according to a user's preferences. That is, the tweets made by the trusted powerful people may be acceptable to the receivers.

On the basis of this interest, we aim to describe human relationships, particularly vertical and hierarchical ones in society. However, our data is restricted to that for people who belong to our university. Although general research is interested in large-sized communities, which community a person belongs to and how he or she is tied to someone else is not clear without reading or analyzing the content of each tweet. Thus, we first compiled tweets made by our restricted group of people. Then, we classified the data into appropriate classes, i.e., certain communities. Finally, we represented the results as a graph and analyzed them.

2 Previous Research

In this section, we give a brief overview of some studies on user classification of Twitter users². We introduce them from two viewpoints in detail: one based on the follower-followed relationship and the other on the topics of tweets.

2.1 Based on Follower-Followed Relationship

Follower-Followed Relationship

In Twitter, a user can subscribe to another person's posts without that person's permission. The procedure for this subscription is called 'follow'. The user's 'follow' action allows him or her to read all the tweets that the followed person makes. This is called 'following', and the 'follower' follows the person to which a subscription was made. In contrast, 'followed' means a person whose posts are read by other users.

Users in Twitter can freely connect to other people because following has no restrictions³. As a result, an enormous number of 'follower-followed' relationships are constructed in the world.

Clustering Based on Modularity

One study simply regarded the follower-followed relationship as a network and attempted to classify the users in the network [1]. This study adopted a benefit object

² All studies reviewed here did not attempt to classify users. We treated the studies as the same type of study because they discussed measures to characterize users and the measures will be possible to extend classification techniques.

³ It is except for users whose twitters are not openly visible, of course.

function Q , i.e., modularity [2], to appropriately divide their university users into clusters as a measure. Q is shown in the following equation.

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) s_i s_j, \quad (1)$$

where m is the total number of edges. A_{ij} is the element of the adjacency matrix, both i and j mean vertices, and k means the degree of the vertices when the network includes n vertices and a pair of nodes $s_i s_j$.

An experiment in the study to find communities found that the precision acquired in the group related to faculties was over 80% in their university. With regard to departments, the precision was over 70%. Thus, the follower-followed relationship enables the network to be divided into existing communities.

Analysis Based on Hyperlink-Induced Topic Search (HITS) Algorithm

Another study also used the follower-followed relationship [3]. This study was based on the hyperlink-induced topic search (HITS) algorithm, which focuses on two values as a measure to describe a user's characteristics: authority and hub [4].

$$Authority(p) = \sum_{v \in S, v \rightarrow p} Hub(v) \quad (2)$$

$$Hub(p) = \sum_{u \in S, p \rightarrow u} Authority(u), \quad (3)$$

where p means a certain page, and ' $a \rightarrow b$ ' shows the link from a to b . Thus, 'Authority' originally depends on the number of links from another page to itself, while 'Hub' depends on those from itself to other pages.

The study extended the idea to a real follower-followed relationship. It showed that some subcategories were detected concerning particular topics, e.g., 'gaming', as a result of an experiment using the modularity.

Furthermore, according to these values, the study anticipated the acquisition of certain groups as shown below, although this hypothesis related to a user's intention was not discussed. This idea, however, seems to have something to do with a user's classification.

1. friendship-wise relationship
2. information seeking
3. information sharing

2.2 Based on Topic

Analysis Based on TwitterRank

A third study adopted the topic-specific measure, i.e., TwitterRank [5]. This study calculated the transition probability matrix, p_t , when a certain topic t was given.

$$P_t(i, j) = \frac{|T_j|}{\sum_{a: s_i \text{ follows } s_a} |T_a|} * sim_t(i, j), \tag{4}$$

where T_j means the number of tweets posted by user S_j , and $\sum_{a: s_i \text{ follows } s_a} |T_a|$ means the total number of posts made by all followers of S_i . Then, $sim_t(i, j)$ in Eq. (4) means the similarity between two users S_i and S_j .

We can easily see that the transition probability p_t depends on the similarity of the topics, although this equation is not explained in detail due to space limitations in this paper. Consequently, the result obtained from the measure, which describes user characteristics by this probability, should differ from the ones obtained from the authority and hub measures.

Clustering Based on Topic

The topics, extracted from the replies between users, were also used in the study we last reviewed [6]. The topic was calculated using categories of nouns based on Wikipedia⁴, the free encyclopedia.

The basic idea of the study is shown in Fig. 1. These sequences of tweets (a reply from R to T) are related to ‘cat’ and ‘dog’, respectively. These tweets are then interpreted as two topics according to the categories in the Wikipedia dictionary: ‘domesticated animals’ and ‘cosmopolitan species’.

Here, the users were considered as having interest in the topics of the tweets. After repeated analysis of this detection of topics, we can see the preferred topics of the users.

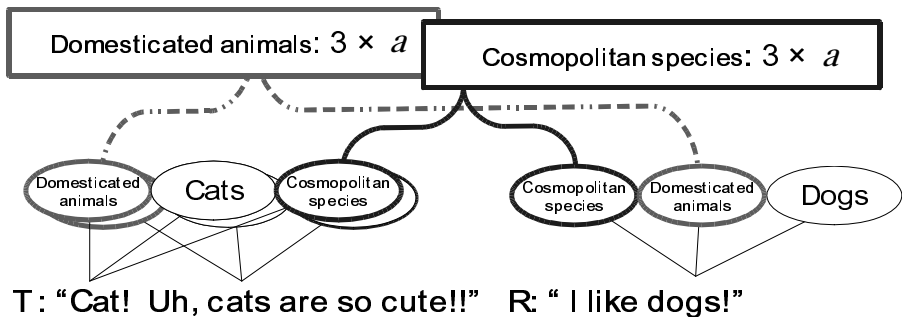


Fig. 1. Sample Topics according to Nouns in Wikipedia

⁴ <http://www.wikipedia.org/>

Afterwards, these data were used in an attempt to classify the users by using the self-organizing map (SOM) algorithm [7]. This seemed to be effective, although the acquired cluster size was small.

The measures in these studies seem to be effective for describing and detecting a user's characteristics, such as preferences. In contrast, they do not clearly represent the user's interpersonal relationships. The first two studies referred to the follower-followed relationship, but this relationship simply indicates a link exists.

For example, in one study [3], this relationship was defined simply by the two measures (authority and hub) being approximately equal to each other. Because this depends on the ratio of two values, it does not mean a real interpersonal relationship between users. In the same way, the studies addressing topics in tweets are valid for detecting a user's interests but are not sufficient for extracting one-on-one relationships.

Therefore, to detect the relationship between users, we focus on certain expressions in tweets, i.e., attitudinal expressions.

3 Proposed Method

As mentioned in the previous section, we focus on using not the follower-followed relationship, but the individual interpersonal relationship shown by expressions used in tweets. In particular, we propose a method for addressing attitudinal expressions.

3.1 How to Extract Hierarchical Information

Attitudinal Expressions

An attitudinal expression is one indicating a speaker's attitude. This expression is defined as "the way that the speaker gives an 'opinion' about his own talk related to his interlocutor" in Ref. [8]. It is a broadly defined term and is capable of explicitly representing our inner state, such as our intentions, emotions, feelings, and so on.

Politeness is one such state, i.e., a respectful feeling towards someone. In Japanese, this type of expression, called an honorific, frequently appears in our daily life as shown in the following example (although all expressions in English have the same surface).

Sensei ga hiru gohan wo otabe-ni-naru.
 (noun + particle) (noun + particle) (verb: exalting expression)
 = My teacher eats lunch.

Watashi ga hiru gohan wo itadaku.
 (noun + particle) (noun + particle) (verb: humbling exp.)
 = I eat lunch.

Watashi ga hiru gohan wo tabe masu .
 (noun + particle) (noun + particle) (verb) (auxiliary verb: polite exp.)
 = I eat lunch.

Japanese honorific expressions are traditionally classified into three classes: exalting, humbling and polite expressions. All the verbs in the examples differ from each other, although these examples are all related to the action ‘to eat’ and include the speaker’s respectful attitude; the first sentence indicates respect toward the subject (ex. teacher) and the second and third sentences indicate respect toward a listener. Moreover, the third sentence includes the marker for a polite expression, i.e., ‘*masu*’.

Extract Hierarchical Information

In the above examples, we can observe the following types of verbs – exalting: ‘*o-tabe-ni-naru*’, humbling: ‘*itadaku*’, and original: ‘*taberu*’. Below is another example sentence.

**Watashi ga hiru gohan wo otabe-ni-naru.*
 (noun + particle) (noun + particle) (verb: exalting expression)
 = I eat lunch.

This example in Japanese is a non-sentence because the speaker uses a verb form indicating respect toward him or herself; this is not ordinary. The verb is usually used for someone else and is not used to express something toward one’s self.

Thus, as seen from this observation, the appearance of an exalting verb enables the interpersonal relationship between the speaker and the subject to be detected; the subject is in a higher position. That is, we can describe the degree of the user’s position and extract the relationship by focusing on such expressions in tweets as a clue for doing so.

3.2 How to Visualize Nodes

When we depict the relationship as a social graph after the classification, the increase of users may result in an ill-formed one because the edges increase so much, the graph becomes complicated [1]. To solve this problem, we must take the visualization of the result into account. Then, we change the degree of the users’ social relationship, which is calculated by the number of linked users, and depict according to the degree in the graph. In particular, we adopt indegree centrality as a measure.

This centrality shows the degree to which a node is linked to other ones. This linkage from the other nodes corresponds to the observation in the previous subsection. That is, the direction in which respect is extended is detected by the use of honorific expressions.

Therefore, by our proposed method, i.e., by focusing on the centrality and changing the node size, we should acquire a more accurate social graph reflected on our real communities.

4 Experiment

Here we describe our experimental procedure.

4.1 Data

Users and Tweets

We treated approximately 400 Twitter accounts in our university. We compiled all messages (indicated by an @ symbol) that the users posted as a reply.

Then, the messages were analyzed and divided into morphemes using Japanese speech tagger MeCab⁵.

Keywords for Honorifics

We regarded the replies including honorific expressions as a target. In addition, in this paper, the expressions are restricted to two polite words, ‘*desu*’ and ‘*masu*’, because these two words have an effect to detect social relationship as well as frequently occurring in our reply data.

Detecting Relationship

The number of honorific replies was counted between all combinations of users. Then, considering each combination, we determined that a user who had a higher number of honorific replies directed at him or her than other users was treated as being in higher position. Afterwards, the detected positions were used for indegree centrality as mentioned in subsection 3.2.

Analyzing Data and Visualizing Tool

We used R⁶ to analyze our data and depict the relationships as a social graph; R is a language and environment for statistical computing and graphics.

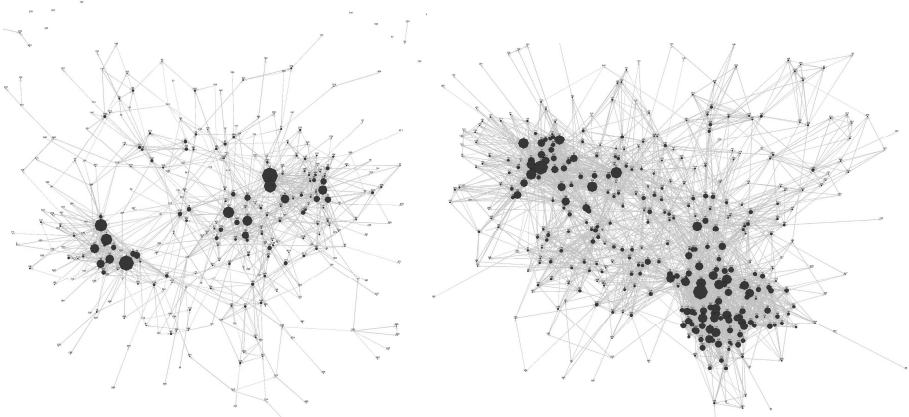


Fig. 2. Acquired Social Graphs: Left: Proposed Method based on Honorific Relationship; Right: Follower-followed Relationship for Comparison. Same Magnification for Both.

Table 1. Number of Nodes and Edges to Depict Each Graph

	Num. of Nodes	Num. of Edges
Proposed Method	326	1,057
Baseline	392	6,098

⁵ <http://mecab.sourceforge.net>

⁶ <http://www.r-project.org>

4.2 Results

Basic Analysis

Two graphs, one based on our proposed method and another for comparison, are shown in Fig. 2. We also show the number of nodes and edges needed for depicting these graphs in Table 1.

Both methods obtained a similar number of nodes, while the number of edges varied greatly between the two methods: one was six times larger. This corresponded with our intuition regarding the number of edges that fewer edges become more obvious in the social graph, in which the edges were sparse on the left, while they were thickly collected on the right. This suggests that the vertical relationship we focused on still remained in the social graph, while horizontal relationships disappeared. As a result, our proposed method produces results that are understandable and superior to the baseline ones based on the follower-followed relationship.

We next discuss the above points from different viewpoints. The magnified graphs are shown in Fig. 3.

We depicted the nodes around the user A as graphs. As mentioned above, one is sparse and the other is thickly collected. Consequently, the existence of user A stands out so clearly that we easily understand his or her social relationship to many users from the left graph. However, in the right graph, the relationships of user A are not clear.

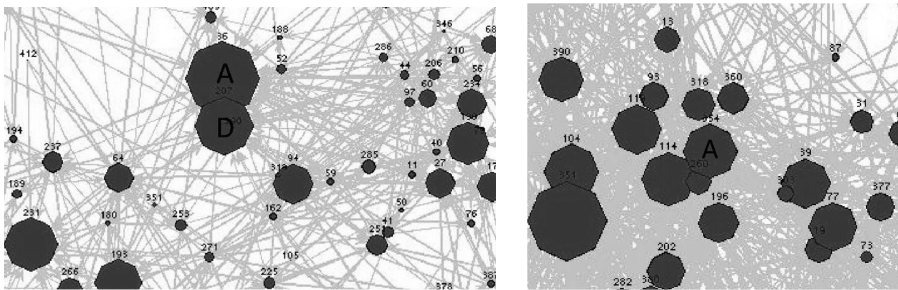


Fig. 3. Magnified Social Graphs around User A: Left: Proposed Method; Right: Baseline for Comparison. Same Magnifications for Both.

Table 2. Indegree Centrality Values Extracted from Top Five Users after Normalization

User	Indegree Centrality	Individual Attribution at Our University
A	1.00	Teaching Staff Member
B	0.94	Teaching Staff Member
C	0.79	Master Program Student
D	0.79	Doctoral Program Student
E	0.72	Master Program Student

This also suggests that our proposed method was effective for describing the user's interpersonal relationships. That is, focusing on honorific expressions is a valid method.

Node Size Depicted in Graph

As mentioned in Section 3, we used the value of indegree centrality to depict the nodes as a graph. We show the top five users whose values are high in Table 2.

The user whose values are higher seemed to be in a higher position socially; the person with the highest value was university staff. Moreover, the top 20 data indicated the same tendency, although only five users are listed in this table due to space limitations.

Comparison of Indegree Centrality

We then compared both methods regarding the indegree centrality. All users whose values were over 0.1 are shown in Table 3.

Each user's value with the proposed method was greater than that of the baseline. This suggests that our proposed method is effective for these users. However, only four users met the conditions. That is, this means that our method had a little effect on some certain users.

Table 3. Indegree Centrality Values in Each Method

User	Indegree Centrality		Individual Attribution at Our University
	Proposed Method	Baseline	
A	1.00	0.68	Teaching Staff Member
B	0.94	0.72	Teaching Staff Member
D	0.79	0.51	Doctoral Program Student
F	0.58	0.32	Teaching Staff Member

This result may depend on the selected expressions because our method processes tweets extracted using two expressions. Many tweets were naturally not selected because they did not contain the expressions. Therefore, investigating other expressions will be our future work.

5 Conclusion

We proposed a method for describing a Twitter user's relationships based on honorific expressions used in tweets. We compiled the tweets and performed an experiment to classify the data. Then, we discussed the results where the depicted social graph was superior to that acquired from the baseline, i.e., by the condition based on the follower-followed relationship.

In this study, we restricted the honorifics to the two auxiliary verbs '*desu*' and '*masu*'. Even with focus on only these two, we could confirm our method was valid. For future work, we need to further discuss the varieties of honorific

expressions or attitudinal expressions, and reveal their effectiveness for indicating societal relationships.

Acknowledgments. This study is supported in part by a Hiroshima City University Grant for Special Academic Research (General Studies 2011). Thank you to everyone concerned.

References

1. Hatamoto, Y., Kurosawa, Y., Mera, K., Takezawa, T.: Clustering Users and their Frequently Posted Words in Micro-blogging Service. In: Proceedings of the 17th Annual Meeting of the Association for Natural Language Processing (2011) (in Japanese)
2. Newman, M.E.J.: Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America* 103(23), 8577–8582 (2006)
3. Java, A., Song, X., Finin, T., Tseng, B.: Why We Twitter: Understanding Microblogging usage and communities. In: Proceeding of the 9th WebKDD and First SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis (2007)
4. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM* 46(5), 604–632 (1999)
5. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: Finding Topic-sensitive Influential Twitterers. In: WSDM 2010: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 261–270 (2010)
6. Kurosawa, Y., Takezawa, T.: Classification of Users and their Posts Focusing on their Reply Action in Micro-blogging Service. In: Proceedings of the 17th Annual Meeting of the Association for Natural Language Processing (2011) (in Japanese)
7. Kohonen, T.: *Self-Organizing Map*, Third Extended edn. Springer, Berlin (2001)
8. Mac, D.K., Auberge, V., Rilliard, A., Castelli, E.: Cross-cultural perception of Vietnamese Audio-Visual prosodic attitudes. In: *Speech Prosody* (2010)

Twitter Sentiment Analysis Based on Writing Style

Hiroshi Maeda, Kazutaka Shimada, and Tsutomu Endo

Department of Artificial Intelligence,
Kyushu Institute of Technology
680-4 Iizuka Fukuoka 820-8502 Japan
{h_maeda, shimada, endo}@pluto.ai.kyutech.ac.jp

Abstract. This paper proposes a new method of sentiment analysis for Twitter. Tweets contain various expressions; e.g., use of emoticons. The usage of these expressions links to the user's identity and individual characters. Handling these characteristics is useful for the sentiment analysis. We focus on writing styles of each user. In this paper, we define three types of writing style; formal and two informal expressions. First, our method classifies each tweet into the three types. Then, it generates classifiers for each writing style. We apply our method to a positive / negative classification task of tweets. In the experiment, the accuracy of our method increased by approximately 3 points as compared with some baseline methods.

Keywords: Sentiment analysis, Positive/negative classification, Writing style.

1 Introduction

Sentiment analysis is one of the hottest topics in natural language processing [7]. There are many target resources for sentiment analysis tasks, such as review documents and blog entries. In this paper, we focus on Twitter. Twitter is one of the most important resources for sentiment analysis tasks. Go et al. [4] have reported a method for positive / negative (PN) classification of tweets with training data acquired by using emoticons. Jiang et al. [5] have proposed a PN classification method using characteristic features in Twitter, such as ReTweet (RT) and reply relations. Brody and Diakopoulos [2] have reported the importance of lengthening as a widespread phenomenon in Twitter. Tweets contain various expressions; e.g., use of emoticons. The usage of these expressions links to the user's identity and individual characters. Rao et al. [8] have proposed a method for classifying latent user attributes; gender, age, regional origin and political orientation. Bar-Haim et al. [1] have reported the importance to identify expert investors for predicting stock price movement from Twitter. Handling these characteristics is useful for many tasks.

Here we focus on writing styles on Twitter. There are various writing styles on Twitter, and they contain different characteristics. For example, tweets written in a colloquial style contain emoticons and characteristic expressions at the end of

Table 1. Examples of each style

Style	Example
Formal	The cherry blossoms are really beautiful at night.
Informal-F	So sweet! I see the lovely cherry blossoms at night :D
Informal-M	Fxxking kewl blossoms at nite!! That's aweeeeeesome!!

Table 2. Subjective definition of each style

Style	Impression
Formal	literary, long sentence
Informal-F	colloquial, soft, sort of femal
Informal-M	colloquial, vulgar, sort of manlike

sentences, such as “!!!!” in the tweet “Too bad!!!!”. They are often related to the polarity of tweets. On the other hand, the polarity of tweets written in a literary style is represented by direct expressions such as lexicons in sentiment dictionaries. Understanding the writing styles is useful for sentiment analysis tasks.

In this paper, we propose a method based on writing styles of tweets for PN classification. The contribution of this paper is to investigate the impact of the writing styles for sentiment analysis. Our method classifies each tweet into three writing styles first. Next it constructs training data for each writing style automatically. Then it generates classifiers for the three writing styles, and classifies an input into positive or negative by using the suitable classifier of the writing style of the input tweet.

2 Writing Style

There are many writing styles on the web. Utilizing the writing style is useful for various natural language processing tasks. Koppel et al. [6] have reported a difference of usage of linguistic expressions in author gender identification. Burger et al. [3] have proposed a method for gender identification on Twitter.

In this paper, we assume that the polarity of words or emoticons depends on the writing style related to author attributes such as gender, age and the Internet community they belong to. For example, the word “school” often contains a negative impact in young people; e.g., “School assignment ... I can’t be bothered.” On the other hand, people in the prime of life sometimes have a positive sentiment in a retrospective look, such as “I had a good time when I was at school.” The emoticon “\ (^ o ^) /” is another instance. This emoticon intuitively expresses the positive sentiment with delight in Japanese. On the other hand, people in the community in 2channel¹ often use this emoticon as a negative expression; e.g., “I’m \ (^ o ^) / screwed.” These phenomena lead to the decrease of the accuracy of sentiment analysis tasks.

To solve this problem, we introduce three writing styles for Twitter sentiment analysis; formal and two informal styles. Table 1 shows examples of each writing

¹ One of the most famous textboards in Japan.

style². It is hard to discriminate the writing style of each tweet objectively and semantically. Therefore, in this paper, the criteria to classify the writing styles are our subjective impressions based on surface expressions of tweets. Table 2 shows the subjective impressions³. The formal style denotes a bookish style. The informal-F style is often used in blogs and e-mails. The informal-M style is more informal than the informal-F style and sometimes uses vulgar expressions⁴. People in the prime of life tend to use the formal style. The two informal styles tend to be used among young people. The usage of them is often distinguished by the net community they belong to.

3 Propose Method

In this section, we explain our proposed method. Figure 1 shows the outline of the method. First, we need to classify tweets into three writing styles described in Section 2. Then, we acquire virtual training data from a corpus by using seed expressions and a sentiment dictionary. For the test phase, we also classify test data into each writing style, and determine the polarity of each tweet by using a suitable classifier for the writing style.

3.1 Writing Style Identification

The writing style identification process consists of two approaches. The basic approach is based on surface expression rules. This approach is used for constructing training data and preparing test data. The second approach is based on a machine learning technique. This approach is used for preparing test data to obtain the high recall rate of the writing style identification process. Finally, we combine the two methods. In other words, we utilize the rule-based method first, and then we identify the writing style of tweets which are not matched with the rules by using the machine learning based method.

Rule-Based Method

The basic identification method is based on surface patterns. It attaches special importance to the precision rate as compared with the recall rate.

There are two types of rules: suffix patterns and patterns in a sentence. Table 3 and Table 4 show examples of surface patterns. Most of suffix patterns are

² Actually, we handle only Japanese tweets in our research. These examples are English translation of Japanese.

³ Note that these words are mere impressions of tweets belonging to each writing style. Classification of tweets to each writing style is based on rules and machine learning using surface expressions. The details will be described in the next section.

⁴ Femal and manlike expressions are a Japanese phenomenon. In Japanese, there are many femal and manlike expressions; e.g., “ore (I)” as a manlike expression and “atashi (I)” as a femal expression. Here the letter in parenthesis is English translation.

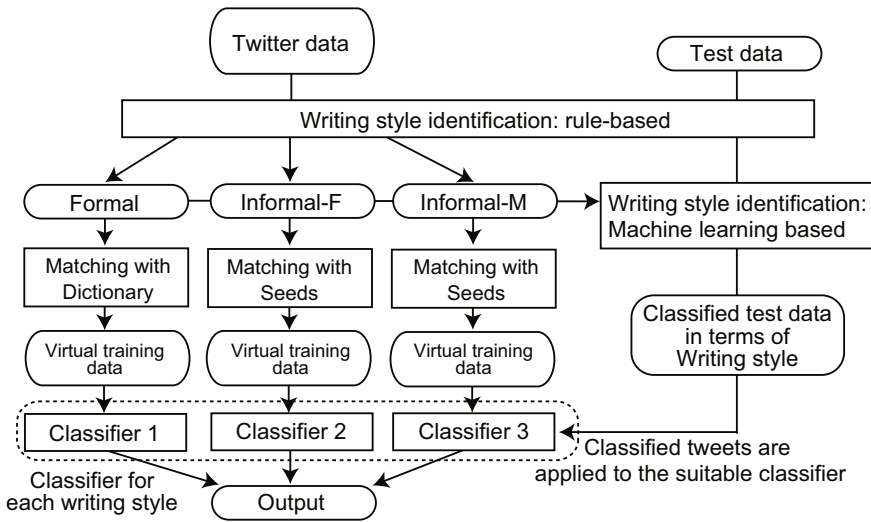


Fig. 1. The outline of our method

Table 3. Suffix patterns

Style	Patterns
Formal	desu, dearu
Informal-F	dayone, damon
Informal-M	kayo, daro

Table 4. Patterns in a sentence

Style	Patterns
Formal	(Nothing)
Informal-F	☆彡, ≧▽≦ (Emoticon)
Informal-M	zamaa(ha-ha!), ore(I)

different expressions of “am/are/is” in English⁵. For the training phase, we use only tweets matched with one of these patterns as training data candidates.

Machine Learning Based Method

Since the rule-based method uses only simple and robust expressions, it is not enough in terms of the recall rate. For test phase, we need a high recall identification method.

To generate the method, we apply a machine learning technique. We call it the ML-based method. We use tweets acquired by the rule-based method as

⁵ Although these expressions contain essentially the same meaning, they have different impressions. “dayone” is a soft expression and “damon” is feminal. On the other hand, “kayo” is fierce and “daro” is manlike.

Table 5. Seed words

Style	Positive	Negative
Informal-F	♪	;)
Informal-M	kitakore (Here it comes!) ossha (boo-yah!)	kuso-sugi (awful) zigoku (hell)

training data for the machine learning. We employ the naive bayes classifier as the method, and letter-level 3-grams as the features. The size of training data is one hundred thousand tweets for each writing style.

3.2 Training Data Acquisition

To generate classifiers suitable to each writing style, we need to construct a large amount of training data. However, collecting training data by hand is costly. In this paper, we apply an automatic training data acquisition with seed words for the informal-F and informal-M styles and use a sentiment lexicon dictionary for the formal style.

First, we explain the method with seed words. Many researchers have proposed methods without rich training data. Riloff and Wiebe [9] have reported a method to learn extraction patterns for subjective expressions. They applied a bootstrapping process to the method. Wiebe and Riloff [12] have proposed a method for creating subjective and objective classifiers from unannotated texts. They used some rules for constructing initial training data. Then they used the data for generating a classifier. Turney [11] has proposed a method for classifying reviews as recommended or not recommended by using some seed words. He used the words “excellent” and “poor” as the seed words, and computed the semantic orientation of a phrase by using the Pointwise Mutual Information (PMI) between the seed word and the phrase. Go et al. [4] also have reported a method with emoticons for the training data acquisition. In this paper we also apply a seed-based method for the informal-F and informal-M styles. Table 5 shows examples of seed words for the two writing styles. For example, in the case the informal-F style, we extract tweets matched with “” from tweets obtained by the rule-based writing style identification, as the virtual positive data. In the same way, we extract tweets matched with “;)”⁶ as the virtual negative data for PN classification. The number of extracted training data was one hundred thousand tweets for the two informal styles.

Next, we describe the method with a sentiment lexicon dictionary for the formal style. We use the dictionary developed by Takamura et al [10]. The lexicons in the dictionary contain the score [-1, +1]. We compute the sum of the polarity scores of each word in each target tweet. Then, we extract tweets if the score is more than a threshold, as the positive data. We also extract tweets if the score is less than a threshold, as the negative data. We investigated the optimal values for the thresholds experimentally. As a result, we set 1.8 to the two thresholds.

⁶ “;)” denotes a part of a crying face or a face with cold sweats in Japanese style.

3.3 PN Classification

On the basis of the extracted training data, we generate classifiers for three writing styles. We apply the naive bayes algorithm to the classifiers. The naive bayes method is a probabilistic model based on Bayes' theorem.

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \quad (1)$$

Here we can use only the numerator of the fraction since the denominator $P(d)$ does not depend on c .

$$\hat{c} = \operatorname{argmax}_c P(c) \prod_{i=1}^n P(x_i|c) \quad (2)$$

where c is the class (P or N) and x_i is a word in a tweet.

As the features, we examine several combinations of letter-level n-grams and word-level n-grams. Our method classifies tweets that are judged by the rule-based and ML-based writing style identification processes.

4 Experiment

We evaluated our method in this section. First, we discuss the accuracy of writing style identification process. We prepared 534 tweets with a writing style tag. The data set consisted of 138 tweets of the formal style, 253 tweets of the informal-F style and 143 tweets of the informal-M style. The criteria of the evaluation were recall, precision and f-measure computed as follows:

$$\begin{aligned} \text{Recall} &= \frac{\# \text{ of tweets classified correctly}}{\# \text{ of tweets of the writing style}} \\ \text{Precision} &= \frac{\# \text{ of tweets classified correctly}}{\# \text{ of tweets classified as the writing style}} \\ \text{F-measure} &= \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \end{aligned}$$

We compared the rule-based method and the method combined with the machine learning approach. Table 6 shows the experimental result. Most of the criteria for the combined method outperformed those for the method with rules only. The weak point of the rule-based method was the recall rate. 144 of 534 tweets were not matched with the rules. They are about 30% of the data set. Hence, the recall rate decreased. This result shows the effectiveness of the combined method with rules and a machine learning technique.

Next, we evaluated our method in terms of PN classification. The number of tweets for the evaluation was 692 tweets. They consisted of 315 positive tweets and 377 negative tweets. We also used approximately one hundred million tweets for the acquisition process of virtual training data. We compared the accuracy

Table 6. Experimental result of writing style identification

		Formal	Informal-F	Informal-M
Rule	Recall	71.7	80.2	21.0
	Precision	77.3	92.7	54.5
	F-measure	74.4	86.0	32.3
Combined	Recall	86.2	89.3	56.6
	Precision	72.1	88.6	71.1
	F-measure	78.5	89.0	63.0

Table 7. Experimental result of PN classification

Method	Accuracy
C(Formal) : baseline1	77.1
C(Informal-F) : baseline2	80.5
C(Informal-M) : baseline3	73.0
Proposed	83.2

rates of our method and some baseline methods. The accuracy rate is computed as follows:

$$Accuracy = \frac{\# \text{ of tweets classified correctly}}{\# \text{ of tweets in the data set}}$$

Table 7 shows the experimental result⁷. The baseline methods in the table denote methods not handling writing style. In other words, the methods generated one classifier from the virtual training data. In the table, “C(Formal)”, “C(Informal-F)” and “C(Informal-M)” are the baseline methods and denote the methods generated from tweets for formal, informal-F and informal-M styles, respectively. For example, the “C(Informal-F)” was a classifier generated from tweets containing virtually-annotated P/N tags by using seed words, “” and “;)” (See Section 3.2). In other words, the “C(Informal-F)” was the “Classifier 2” in Figure 1. On the other hand, the accuracy of the proposed method was the average accuracy of three classifiers which were suitable for each writing style⁸. The proposed method outperformed the baseline methods.

Table 8. Result of PN classification in detail

Method	C(Formal)	C(Informal-F)	C(Informal-M)
Formal (240)	77.1*	75.0	72.9
Informal-F (307)	78.0	88.3*	71.0
Informal-M (145)	78.6	73.1	82.8*

⁷ In this experiment, the best features for formal, informal-F and informal-M were letter-level 4-grams, letter-level 3-grams and letter-level 4-grams, respectively.

⁸ Actually, the accuracy was the weighted average value based on the number of tweets belonging to each writing style because the tweets of each writing style were not uniform distribution. In other words, it was the micro average

Table 8 shows the detail of the experimental result. In the table, “Formal (240)” denotes that the number of tweets that were classified as “formal style” by rule and ML-based identification processes was 240 tweets from 692 tweets in the test data. In other words, the numbers of tweets of the formal, informal-F and informal-M styles were 240 tweets, 307 tweets and 145 tweets⁹, respectively. The accuracy of the informal-F style classifier for tweets that were identified as the informal-F style was 88.3%. The values with “*” in the table denote the accuracy rates of our method, which classified tweets by using suitable writing style classifiers. The micro average of these values was 83.2% in Table 7. On the other hand, the accuracy was 75.0 % in the case that the informal-F style classifier classified tweets that were identified as the formal style. In a similar way, 73.1% in the second column was the accuracy of tweets that were identified as the formal style by the informal-F style classifier. Therefore, the micro average of values in the second column, namely 75.0, 88.3 and 73.1, was the accuracy of the “C(Informal-F) : baseline2” in Table 7. The accuracy rates of our method outperformed those of non-suitable classifiers. For example, the accuracy of the formal tweets by the formal style classifier produced the best performance (77.1%) as compared with the formal tweets by the informal-F classifier (75.0%) and the informal-M classifier (72.9%). The other writing style had the same tendency (88.3 vs. 78.0 and 71.0 for Informal-F and 82.8 vs. 78.6 and 73.1 for Informal-M). This result shows the effectiveness of classifiers considering the writing styles for the PN classification.

Other evaluation criteria were the recall, precision and F-measure, which were also used for the writing style identification. Table 9 shows the result of this experiment in terms of the evaluation criteria. The Recall (Pos), Precision (Pos) and F-measure (Pos) denote the recall, precision and F-measure for positive tweets. In the same manner, the Recall (Neg), Precision (Neg) and F-measure (Neg) denote the recall, precision and F-measure for negative tweets. Table 9 contains three tables; the results for formal tweets, informal-F tweets and informal-M tweets. In each table, “*” has the same meaning at Table 8, i.e., the results of our method, which classified tweets by using suitable writing style classifiers. Most of the values by the proposed method were higher than those by the methods, namely non-suitable classifiers. Our method outperformed the non-suitable classifiers on all the F-measure values. This result also shows the effectiveness of our method.

In our method, the writing style identification process is one of the most important parts. If the process generates misclassification, it leads to the decrease of the PN classification accuracy. For the test data, this process was performed automatically with the rule-based and ML-based methods. However, the precision rate of the combined method was not always high. The average of F-measure computed from Table 6 was approximately 75%. On the other hand, the accuracy

⁹ Note that the identification results were based on the method in Section 3.1. The input tweets of PN classification were not human annotated data but actual result of our method.

Table 9. Recall, precision and F-measure of PN classification

Formal tweets			
	C(Formal)	C(Informal-F)	C(Informal-M)
Recall (Pos)	69.3*	54.5	46.5
Precision (Pos)	74.5*	79.7	81.0
F-measure (Pos)	71.8*	64.7	59.1
Recall (Neg)	82.7*	89.9	92.1
Precision (Neg)	78.8*	73.1	70.3
F-measure (Neg)	80.7*	80.6	79.8
Informal-F tweets			
	C(Formal)	C(Informal-F)	C(Informal-M)
Recall (Pos)	73.8	80.6*	55.8
Precision (Pos)	81.9	96.3*	80.3
F-measure (Pos)	77.6	87.8*	67.9
Recall (Neg)	82.3	96.6*	84.4
Precision (Neg)	74.2	82.1*	65.3
F-measure (Neg)	78.1	88.7*	73.6
Informal-M tweets			
	C(Formal)	C(Informal-F)	C(Informal-M)
Recall (Pos)	66.7	38.9	74.1*
Precision (Pos)	73.5	77.8	78.4*
F-measure (Pos)	69.9	51.4	76.2*
Recall (Neg)	85.7	93.4	87.9*
Precision (Neg)	81.3	72.0	85.1*
F-measure (Neg)	83.4	81.3	86.5*

with the rule-based method was approximately 85%¹⁰. Therefore, we evaluated our method with tweets that were identified by only the rule-based method, namely high confident tweets. As a result, we deleted 236 of 692 tweets, i.e., the test data consisted of 456 tweets. In this situation, the accuracy of our method was shown a rise to 86.4%. On the other hand, the best accuracy of the baseline methods was 83.8% by the informal-F style classifier. This result shows the significance of the writing style identification for the PN classification.

5 Conclusions

This paper proposed a new method focusing on writing styles for Twitter sentiment analysis. We introduced three types of writing style; formal and two infor-

¹⁰ The accuracy was based on tweets matched with rules, namely 390 of 534 tweets. The accuracy is computed by $\frac{\# \text{ of correct tweets}}{\# \text{ of tweets matched with rules}}$. In other words, the rule-based method generates high accuracy under the condition that we do not treat tweets which are not matched with the rules. Note that this value can not be computed from Table 6.

- [9] Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP 2003 (2003)
- [10] Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientations of words using spin model. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 133–140 (2005)
- [11] Turney, P.D.: Thumbs up? or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 417–424 (2002)
- [12] Wiebe, J., Riloff, E.: Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In: Gelbukh, A. (ed.) CILing 2005. LNCS, vol. 3406, pp. 486–497. Springer, Heidelberg (2005)

Extraction of User Opinions by Adjective-Context Co-clustering for Game Review Texts

Kevin Raison, Noriko Tomuro, Steve Lytinen, and Jose P. Zagal

DePaul University, College of Digital Media,
243 S. Wabash, Chicago, IL 60604 USA
raison@chatsubo.net, {tomuro,lytinen}@cs.depaul.edu,
jzagal@cdm.depaul.edu

Abstract. We present our preliminary work on extracting fine-grained user opinions from game review texts. In sentiment analysis, user-generated texts such as blogs, comments and reviews are usually represented by the words which appeared in the texts. However, for complex multi-faceted objects such as games, single words are not sufficient to represent opinions on individual aspects of the object. We propose to represent such an object by pairs of aspect and each aspect's quality/value, for example "great-graphics". We used a large adjective-context co-occurrence matrix extracted from user reviews posted at a game site, and applied co-clustering to reduce the dimensions of the matrix. The derived co-clusters are pairs of row clusters \times column clusters. By examining the derived co-clusters, we were able to discover the aspects and their qualities which the users care about strongly in games.

Keywords: adjective-context relation, game reviews, co-clustering, sentiment analysis, opinion mining.

1 Introduction

Sentiment analysis has been receiving increased attention in recent years in Natural Language Processing (NLP) [7]. With the proliferation of user posts on online social media, such as weblogs, product reviews and message boards, NLP techniques have been applied to automatically extract and analyze user opinions and sentiment expressed in those texts. Sentiment analysis has been incorporated in a variety of applications, for example in obtaining product marketing information [8], tracking political opinions [10], and searching for "buzz" (hot topics) in social networks. Games, especially videogames, are one of the domains which would benefit from automated sentiment analysis. Not only is there an enormous amount of data (reviews, comments, etc.) available already (for instance at game sites such as Gamespot (www.gamespot.com), IGN (www.ign.com) and Giantbomb (www.giantbomb.com)), new games are created continuously and rapidly. Moreover, game users/players are generally quite passionate and vocal about the games they like or dislike, therefore reviews tend to be long.

In sentiment analysis (or opinion mining), user-written texts are typically represented by the words which appear in them, then categorized for polarity orientation (positive/negative/neutral) or emotions (e.g. joy, sadness). However, for complex objects that have many facets/aspects such as games, single words are not sufficient to represent opinions on individual aspects of the object. For example, reviews such as “graphics are great, but gameplay is horrible” and “graphics are horrible, but gameplay is great” are critically indistinguishable if only single words are used (represented by the same four single words: “graphics”, “great”, “gameplay”, “horrible”). A better approach is to represent each aspect and its quality/value together, for instance “great-graphics” and “horrible-graphics”. This representation scheme can also express more fine-grained as well as accurate opinions as conveyed in the texts.

In this paper, we present the preliminary results of our work on extracting fine-grained opinions from game review texts. We started with the adjective co-occurrence dataset used in [11,12], which contained the adjective bigrams (i.e., bigrams in which either word is an adjective) extracted from the user reviews posted at a game site (Gamespot). In this dataset, the adjectives and the *context words* (i.e., words which co-occurred with the adjectives in a bigram) were represented separately, by a matrix of adjectives (on the rows) \times context words (on the columns). Then we applied co-clustering [3] to reduce the matrix. Reducing the size of the matrix was necessary in order to make the computation feasible, but it also served as a way to group similar words (e.g. “graphic”, “look”) or typographical errors (e.g. “graphic”, “grafic”). Co-clustering is a technique which clusters the rows and columns of a matrix simultaneously while preserving the dependencies between them. Then we examined the derived *co-clusters* (manually) to discover aspects of games and their qualities which users care about strongly. We also used those co-clusters to cluster games and investigated the results for any interesting patterns that might have emerged.

2 Related Work

Co-clustering has been used in several previous works to capture the dependency between two variables (or objects and features), which are typically represented by the rows and columns of a matrix. For example, [11,12] applied co-clustering in the task of document categorization, and showed improved results obtained by utilizing the set of document clusters produced by co-clustering as compared to the clusters generated by clustering on a single dimension. Another study, [4], applied co-clustering in generating product recommendations. This study represented the item rating scores posted by the users by a matrix of items on the rows and users on the columns, and applied co-clustering to discover the patterns of user preferences. Then they used the derived co-clusters to predict rating scores and recommendations for new users.

Since the early stages of work on sentiment analysis in NLP, adjectives have been effectively used to extract user opinions or polarity from texts [6,7]. Recently, several works have used adjectives and the nouns which co-occurred with

them (e.g. “wonderful ideas”, “horrible taste”) for the purpose of extracting more accurate or fine-grained opinions. For example, [9] extracted adjective-noun pairs, which are in the dependency/modifying relation, from the opinions posted at an online forum on eGovernment for the purpose of mining public opinions on various government decisions; while [1] used a similar approach to identify consumer preferences from product reviews (posted at Amazon).

The work which is closest to ours would be [13]. They extracted adjective-noun dependency pairs from parsed user review texts, and applied a model based on Latent Dirichlet Allocation (LDA) to derive clusters of adjective-noun pairs (where each cluster is a pair of an adjective set and a noun set). Whereas in our work, we used bigrams surrounding adjectives, and applied the *information-theoretic* co-clustering algorithm from [1] to derive co-clusters (of adjective-context pairs).

3 Game Review Dataset

We conducted experiments using the dataset from [11,12]. This dataset contained 723 adjectives and 5,000 context words (which appeared in a bigram surrounding an adjective, i.e., one word before and one word after the adjective). The data was extracted from the corpus of user reviews posted at Gamespot (as of April, 2009). The corpus covered 8,279 game titles, and the entries in the data were raw frequencies of the bigrams/co-occurrences in the corpus.

Were we to construct the adjective-context co-occurrence matrix from this dataset without compression, we would end up with a total of 3,615,000 ($= 723 \times 5,000$) pairs – prohibitive to use in any computational task. Although around half of the entries (48%) had a value of zero, the frequency distribution had a rather fat tail: the percentages of the entries with values ≤ 2 , ≤ 3 and ≤ 4 (all including 0) were 85%, 89% and 91% respectively. That means we will still be left with over 300,000 ($\approx 9\%$) features, even if we were to keep the ones with frequencies ≥ 5 . Dimensionality reduction was obviously necessary.

4 Co-clustering

To reduce dimensions of the data, we first applied a co-clustering algorithm. In general, a co-clustering algorithm works such that, given a matrix of m rows and n columns (i.e., $m \times n$), it generates p row clusters and q column clusters (where $p \leq m$ and $q \leq n$) by exploiting the mutual dependency (or *duality*) between the rows and the columns. The particular algorithm we used in this work is *information-theoretic* co-clustering as described in [1], which reduces the dimensions of rows and columns simultaneously while minimizing the loss of Mutual Information (MI) contained in the matrix. Formally, the MI between two random variables x and y , denoted $I(x, y)$, is:

$$I(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

where $p(x, y)$ is the joint probability of x and y , and $p(x)$ and $p(y)$ are the (marginal) probability of x and y respectively. MI is symmetric, and indicates the mutual dependence between two random variables. $I(x, y) = 0$ if x and y are independent, or non-zero (positive or negative) otherwise. In our case, $I(x, y)$ essentially indicates how well a context word is correlated with a given adjective, and vice versa. Thus by applying co-clustering to our data, we can identify the aspects and qualities of games that the users care about and which might be different from other products or domains.

5 Results and Discussions

5.1 Adjective/Context Co-clusters

In the current work, we generated 100 row/context clusters and 30 column/adjective clusters (thus a total of 3,000 co-clusters). We chose those numbers of clusters after experimenting with various configurations and choosing the one which we thought gave the least tradeoff between the loss in MI and the reduction of the dimensions.

There were a number of interesting aspects to the co-clusters we obtained. First on the individual clusters in the two sets (adjective and context) of clusters. We observed that, in both sets, some clusters consisted of largely general domain-independent words, while others were made of rather domain-specific words, as expected. For example, some adjective clusters consisted of general evaluation modifiers that apply to any domain (e.g. {great, amazing, excellent, fantastic, incredible,...}, {good, perfect, terrible, mediocre, awful, superior,...}), or speed/pace modifiers (e.g. {fast, slow, quick, sluggish, active, steady, swift,...}). Whereas the examples of clusters which contained game-specific words would include an adjective cluster: {online, cooperative, offline, endless, competitive, fantasy, wireless,...}, and context clusters: {franchise, installment, released, changed, purchase, brought, legend, remake,...} and {combat, fighting, puzzle, shooting, result, deliver, platforming, tactical, hack, strategic,...}.

Another observation was that some adjectives which are used frequently in the game domain were clustered together with other adjectives which indicate sentiment, for example {addictive, fun, exciting, enjoyable, entertaining, challenging, rewarding,...} and {repetitive, boring, frustrating, tedious, bland, annoying, uninspired,...}. Although our clustering method did not incorporate any specific parameter to differentiate the sentiment polarity, it turns out that most adjective clusters consisted of mostly either positive or negative words, thus we were able to obtain a rough idea on the sentiment of those adjectives (e.g. “addictive” is positive; “repetitive” is negative). However, further research is necessary to do a detailed sentiment analysis of game reviews, which we plan to do in our future work.

Yet another observation was that many various misspellings of a word were grouped in the same cluster, as we had hoped. In addition, synonyms of a word were often included in the same cluster with the misspellings as well, thereby making the clustering even more effective. For example, the following clusters were produced:

- multiplayer, multilayer, mutiplayer, multyplayer, mutliplayer, muliplayer, multilayer, multiplay, playin, multi, community, coop, buddy, co
- graphic, grafic, grafix, grahpic, graphix, grapic, graph, cg, cgi, fx, gfx, hd, hdtv, look, looked, looking, lookin
- character, charachter, charactor, charater, charcter, charecter, chracter, char, antagonist, protagonist, villain, villian

Next on the co-clusters (where each co-cluster is a pair of an adjective cluster and a context cluster). Table 1 shows some examples of notable, high frequency co-clusters. We have observed that many high frequency co-clusters had context clusters which refer to the overall look and feel of games, for instance {graphics, look, sound, music,..} and {game, overall, line, conclusion, qualities,..} – which suggests that users care strongly about the aesthetics of the game and gameplay. On the other hand, co-clusters with context clusters containing concrete objects such as {map, gun, city, step, explosion, planet,..} had relatively low frequencies – implying that specific objects/props used in the game might not be so important.

We also further examined the co-clusters for high frequency adjective-context word pairs, as listed in Table 2. While “good”, “great” and “bad” are general, high-level adjectives, “fun” and “pretty” are not. Having “fun game” as the 3rd-most frequent word pair suggests that it is an extremely important thing in the game domain. “Fun play” and “pretty game” are also high in the frequency as well. Other notable pairs include:

Table 1. Some Notable Adjective-Context Co-clusters (Based on Frequency)

Frequency		Words
50,845	Adj	great, amazing, excellent, fantastic, incredible, decent, outstanding, brilliant, wonderful, flawless, stunning,
	Context	graphics, look, sound, music, voice, idea, job, soundtrack, acting, audio, physic, lighting, actor, storie,
42,502	Adj	basic, simple, smooth, innovative, interesting, linear, immersive, engaging, balanced, fluid, compelling, familiar,
	Context	game, overall, line, conclusion, qualities, summary, theory, row, equal, bioshock, unleashed
27,382	Adj	new, original, total, disappointing, additional, updated, expanded, popular, retro, freestyle, successful, acclaimed
	Context	bit, whole, unlock, content, brand, introduce, dimension, disaster, table, awkward, dated, memory,
17,189	Adj	bad, hard, difficult, stupid, tough, mean, smart, sad, tricky, impossible, lagging, negative, screwed,
	Context	control, kill, pick, learn, task, pull, achieve, navigate, maneuver, repeat, execute, achieve, navigate, solve,
5,796	Adj	unique, realistic, arcade, cinematic, sandbox, polished, enhanced, revamped, authentic, novel, reminiscent,
	Context	effect, animation, visual, design, texture, presentation, cut, art, cinematic, scenery, blend, display, script,

Table 2. High Frequency Adjective-Context Word Pairs

Pairs	Frequency
good:game	6,029
great:game	4,956
fun:game	4,811
good:graphics	4,688
good:reviews	4,664
pretty:good	4,440
new:releases	4,374
new:top	4,323
bad:game	3,974
good:thing	3,746
single:player	3,734
fun:play	3,668
overall:game	3,658
good:look	3,623
good:sound	3,556
good:games	3,527
pretty:game	3,512

- “good reviews” – it seems having/receiving good reviews are very important
- “new releases” – staying on the cutting edge also seems important in this domain
- “good sound”, “good graphics”, “pretty game” and “good look” – the audio/visual seems highly important to users

Furthermore, we calculated the Mutual Information for each co-cluster (i.e., MI between an adjective cluster and a context cluster)¹ Unlike frequency, MI is based on probability, and indicates the degree of (positive/negative) dependency beyond coincidence, therefore would be able to give us hints on the linguistic characteristics, in particular the lexical associations in our case, particular to this domain that were not captured by frequency. For the purpose of this paper, of the 3,000 co-clusters we focused on some of those which express sentiment.

Table 3 shows some examples. In the calculated MI scores, the adjective cluster {addictive, fun, exciting,...} had a notable high association with the context cluster {interesting, unique, deep, innovative,..}, suggesting that games which are addictive/fun/exciting tend to have those attributes as well (based on the users’ perception)² On the other hand, the adjective cluster {repetitive, boring,

¹ To clarify, the co-clustering algorithm we used (described in the previous section) computed MI between words (i.e., an adjective and a context word). Here, we computed MI between clusters, which were derived by the co-clustering algorithm.

² Note that there were two other context clusters which had a higher MI score with this adjective cluster. However, words in those context clusters were quite common in general texts (thus not particular to the game domain), therefore not reported here. Other examples presented in Table 3 were the same way.

frustrating,..} was strongly associated with the context cluster {hour, long,..}, suggesting that time length (especially the waiting time) might be a critical attribute of a game which the users feel annoying (maybe more than software/game glitches). Another example, from the perspective of context words, is the cluster {problem, flaw, issue..}. This cluster was strongly associated with the adjective cluster {core, standard, fundamental,..}, suggesting that users are writing about (what they think are) serious problems about the given game in the review – probably for the purpose of sharing the information and their opinions with other users.

Table 3. Some Notable Adjective-Context Co-clusters (Based on Mutual Information)

MI		Words
1.6047949	Adj	addictive, fun, exciting, enjoyable, entertaining, challenging, rewarding, super, exhausting, replayable, approachable,...
	Context	interesting, unique, deep, innovative, immersive, engaging, creative, pure, cute, compelling, dynamic, interactive, diverse,...
1.1835389	Adj	repetitive, boring, frustrating, dull, tedious, bland, annoying, stale, uninspired, monotonous, confusing, irritating,...
	Context	hour, long, week, waiting, hooked, frequent, min, lasted, entertain, forgotten, stopped, anticipated, hr, longest, dragged,...
3.8537414	Adj	core, standard, fundamental, serious, big, minor, raw, friendly, hardcore, major, common, drastic, significant, inducing,...
	Context	problem, flaw, issue, difference, complaint, disappointment, mistake, gripe, fault, pain, letdown, downside, plus, drawback,...

5.2 Game Clusters

Next, we represented the games using the derived 3,000 co-clusters as features and clustered all 8,279 games, as well as several subsets of them segmented by game platform types. Games which were grouped in the same cluster by these processes should share a similar pattern of characteristics on the aspects which the users strongly care about, for example games which are “overall-innovative”, and have “great-graphics” but “difficult-control”. To cluster games, we applied the standard K-means algorithm.

Given the subjective nature of the data corpus, we were unsure of which values for K (the number of clusters) would be the most productive and/or interesting. We hypothesized that certain games, especially those that were described in similar terms, would perhaps tend to group together, forming clusters that remained stable for varying values of K. We also imagined that, given the wide variety of games, more clusters might prove more valuable allowing us a more nuanced understanding and view.

Table 4 shows some example clusters of PS3 games when K=30 was used. For lower values of K (e.g. 30, 50), we noticed strong clusters for sports games. This was particularly noticeable for sports games based on highly popular professional league sports (e.g. American football, basketball, hockey, and soccer) [see

Table 4. Example PS3 Game Clusters ($K = 30$)

Cluster	Games
27	PS3:ProEvolutionSoccer2009, PS3:ProEvolutionSoccer2008, PS3:NHL09, PS3:NHL08, PS3:NCAAFootball09, PS3:NCAAFootball08, PS3:NBA2K9, PS3:NBA2K8, PS3:NBA2K7, PS3:MajorLeagueBaseball2K7, PS3:MaddenNFL09, PS3:MaddenNFL08, PS3:MaddenNFL07, PS3:MLB09TheShow, PS3:MLB08TheShow, PS3:FIFASoccer09, PS3:FIFASoccer08, PS3:All-ProFootball2K8
22	PS3:WWEsmackDownvs.Raw2009, PS3:WWEsmackDown!vs.RAW2008, PS3:TonyHawk'sProvingGround, PS3:TonyHawk'sProject8, PS3:NeedforSpeedProStreet, PS3:MLB07TheShow, PS3:GuitarHeroWorldTour, PS3:DragonBallZBurstLimit, PS3:ArmoredCore4
15	PS3:TheGoldenCompass, PS3:TheChroniclesofNarniaPrinceCaspian, PS3:RockRevolution, PS3:OverlordRaisingHell, PS3:HellboyTheScienceofEvil, PS3:Beowulf

Table 4, Cluster 27]. However, there were also noticeable clusters of games that featured what we called extreme sports (e.g. wrestling, skateboarding) [see Table 4, Cluster 22]. As we increased K , we noticed that the clusters for sports titles, while generally stable, began to differentiate themselves by sport. So, one cluster would now feature baseball and hockey games, while another only basketball games. This clustering for low values of K is interesting because, other than the fact that these games simulate a real-life sport, the sports in and of themselves do not necessarily have that much in common with each other (and, arguably, their videogame representations are not that similar either). However, we did notice that this segmentation also seemed to mirror some of Kayali and Purgathofers [14] categories for sports games. They describe extreme sports games as those that do “not exclusively signify the simulation of extreme sports like skydiving [...] but an exaggerated vision of any sport.” With higher values of K , an over-the-top snowboarding game, such as *SSX*, might appear together with a less-realistic soccer game such as *FIFA Street*, rather than the more simulation-based *FIFA* [see Table 5, Cluster 10]. Similarly, deep sports simulation games such as *MLB Front Office Manager*, *Season Ticket Football 2003* and *Championship Manager 2007* appeared together [see Table 5, Cluster 7]. These are all games that focus on “in-depth team management, multi-season play and a flurry of statistical data” in which “active gameplay [...] may even be excluded” [14].

Table 5 shows example clusters of all games when $K=500$ was used. It would seem that the logic behind the clustering reflects differences and styles in gameplay. To name a few examples outside of sports games, we found clusters that grouped different flavors of shooting games (e.g. tactical, first-person), driving, and fighting games. When $K=500$, we found clusters of games that were not only very similar in gameplay, but additionally belonged to the same franchise such as *Sonic the Hedgehog* games [see Table 5, Cluster 8] or the survival-horror games in the *Silent Hill* series [see Table 5, Cluster 6]. Unfortunately, it is not that simple. Not all clusters can be understood due to gameplay reasons. For example, some clusters grouped games that were tie-ins to Hollywood releases [see Table 4, Cluster 15] such as *The Golden Compass*, *The Chronicles of Narnia: Prince*

Table 5. Example Game Clusters (K = 500)

Cluster	Games
1	ds:TheLegendofZeldaPhantomHourglass, ds:NewSuperMarioBros, Xbox360:NinjaGaidenII, Xbox360:MassEffect, Wii:SuperMarioGalaxy, PS3:MetalGearSolid4GunssofthePatriots, PS2:FinalFantasyXII, PS2:DragonQuestVIII
2	ds:MetroidPrimeHunters, Xbox360:Quake4, Xbox360:Halo3, Xbox360:GearsOfWar, PS3:UnrealTournament3, PS3:TomClancy'sRainbowSixVegas, PS2:Killzone, PC:StarWarsEmpireatWar, PC:AgeofEmpiresIII
3	ds:MarioHoops3on3, Xbox360:WWESmackDownvs.Raw2009, Xbox360:NHL08, Xbox:FIFASoccer06, Xbox360:MaddenNFL09, Wii:RockBand, PSP:RidgeRacer, PS3:TonyHawk'sProject8, PS3:NBA2K8, PS2:TonyHawk'sProSkater3
4	ds:PrinceofPersiaTheFallenKing, Xbox:FindingNemo, Wii:LooneyTunesAcmeArsenal, PS2:CuriousGeorge, PS2:Disney'sTarzanUntamed, PS2:HarryPotterandtheSorcerer'sStone, GameBoyAdvance:TheChroniclesofNarniaTheLion, TheWitchandTheWardrobe
5	ds:MajorLeagueBaseball2K7, Xbox360:NCAAMarchMadness08, Xbox360:NHL2K9, Xbox:UEFAEuro2004, Xbox:TonyHawk'sProSkater4, Xbox:TigerWoodsPGATour07, Wii:ProEvolutionSoccer2008, PSP:MaddenNFL08, PS2:NASCARTThunder2004
6	Xbox:SilentHill4TheRoom, Xbox:SilentHill2RestlessDreams, PS2:SilentHillOrigins, PS2:Obscure, PC:SilentHill4TheRoom, PC:SilentHill3, PC:SilentHill2
7	Xbox360:MLBFrontOfficeManager, PS3:MLBFrontOfficeManager, PS2:ChampionshipManager2007, PC:SeasonTicketFootball2003, PC:MLBFrontOfficeManager, GameBoyAdvance:FIFAWorldCupGermany2006
8	Xbox:ShadowtheHedgehog, Wii:SonicUnleashed, Wii:SonicRidersZeroGravity, PSP:SonicRivals, PSP:Ratchet&ClankSizeMatters, PS3:SonictheHedgehog, PS2:SonicHeroes, PS2:ShadowtheHedgehog, GameCube:SonicHeroes, GameCube:SonicGemsCollection, GameCube:SonicAdventureDXDirector'sCut
9	Xbox:Wallace&GromitCurseoftheWere-Rabbit, Xbox:CuriousGeorge, Xbox:AvatarTheLastAirbender, PS2:Wallace&GromitCurseoftheWere-Rabbit, PS2:AvatarTheLastAirbender, PC:CuriousGeorge, GameCube:AvatarTheLastAirbender
10	ds:MaddenNFL2005, Xbox360:MXvs.ATVUntamed, Xbox:WWFRaw, Xbox:SSX3, Xbox:AmpedFreestyleSnowboarding, Xbox:Amped2, WiiVirtualConsole:MarioKart64, WiiVirtualConsole:F-ZeroX, Wii:WWESmackDown!vs.RAW2008, PSP:VirtuaTennisWorldTour, PSP:NFLStreet2Unleashed, PSP:HotShotsGolfOpenTee, PSP:FlatOutHeadOn, PSP:DragonBallZShinBudokai-AnotherRoad, PS3:TopSpin3, PS3:TNAiMPACT!, PS2:WWFSmackDown!JustBringIt, PS2:WWESmackDownvs.Raw2009, PS2:WWESmackDown!ShutYourMouth, PS2:TNAiMPACT!, PS2:SegaSportsTennis, PS2:SSX, PS2:HotShotsGolfFore!, PS2:FireProWrestlingReturns, PS2:FIFAStrreet, PS2:DefJamVendetta, PS2:BackyardWrestling2ThereGoestheNeighborhood, PS2:ArenaFootball, PS2:ATVOffroadFury, PC:TopSpin2, PC:TigerWoodsPGATour06, PC:SensibleSoccer2006, PC:Crashday, GameCube:WaveRaceBlueStorm, GameCube:WWEWrestleManiaXIX, GameCube:SuperMonkeyBall, GameCube:SSX3, GameCube:MajorLeagueBaseball2K6, GameCube:1080Avalanche, GameBoyAdvance:MarioTennisPowerTour, GameBoyAdvance:MarioGolfAdvanceTour, GameBoyAdvance:DragonBallZSupersonicWarriors

Caspian and *Beowulf*. Other games in this cluster, although not tied to movies, also reflected similar fantasy and fictive elements. In another example, we found several games based on animated television series such as *Curious George* and *Avatar: The Last Airbender* [see Table 5, Cluster 9]. It is possible though that, due to their “tie-in” nature, many of these titles are less-notable or interesting in gameplay terms thus allowing for greater saliency or importance of what seems like a secondary characteristic (being based on a cartoon or movie).

6 Conclusions and Future Work

In this paper, we presented the results of extracting user opinions by applying co-clustering to an adjective-context co-occurrence matrix. From the derived co-clusters, we discovered that game users tend to care about the overall look and feel aspect of the game more than concrete elements used in the game. The results also indicated that, based on the word frequency, “fun” is an extremely important element of a game as well. However, these results are still preliminary; further investigation is needed to obtain results which could provide more, and different insights and discoveries about user preferences in the game domain.

In future work, we plan to do more thorough and deeper analysis of the linguistic characteristics of the game domain in conjunction with user sentiment. In the current work, we discovered some indications of sentiment for words such as “addictive” and “repetitive”, but the justification was not rigorous. In the next work, we plan to incorporate sentiment explicitly in the analysis, possibly through review rating scores, and investigate the language of the game domain.

Techniques for dimensionality reduction need more experimentation as well. Other techniques we are considering of comparing with the current work include Multi-Dimensional Scaling, graph-based algorithms, and those which derive latent dimensions/factors such as LDA [13].

Game clustering also needs much more work. In addition to further experimentation with clustering (by using other algorithms, also with various parameter values), we plan to conduct rigorous evaluation of both qualitative and quantitative results. Finally, as mentioned earlier, we are planning to develop a game recommender system using the derived game clusters, and test the system with real users.

References

1. Archakn, N., Ghosea, A., Ipeirotis, P.: Show me the Money! Deriving the Pricing Power of Product Features by Mining Consumer Reviews. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007), pp. 56–65 (2007)
2. Bisson, G., Hussain, F.: Chi-Sim: A New Similarity Measure for the Co-clustering Task. In: Proceeding of the 7th International Conference on Machine Learning Applications (ICMLA 2008), pp. 211–217 (2008)
3. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-Theoretic Co-clustering. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 89–98 (2003)
4. George, T., Merugu, S.: A Scalable Collaborative Filtering Framework Based on Co-clustering. In: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), pp. 625–628 (2005)
5. Hartigan, J.A.: Direct Clustering of a Data Matrix. *Journal of the American Statistical Association* 67(337), 123–129 (1972)
6. Hatzivassiloglou, V., McKeown, K.: Predicting the semantic orientation of adjectives. In: Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics (EACL 1997), pp. 174–181 (1997)

7. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2), 1–135 (2008)
8. Popescu, A., Etzioni, O.: Extracting Product Features and Opinions from Reviews. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, pp. 339–346 (2005)
9. Stylios, G., Christodoulakis, D., Besharat, J., Kotrotsos, I., Koumpouri, A., Stamou, S.: Public Opinion Mining for Governmental Decisions. *Electronic Journal of Electronic Government* 8(2), 202–213 (2011)
10. Thomas, M., Pang, B., Lee, L.: Get out of the vote: Determining support or opposition from Congressional Floor-Debate Transcripts. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pp. 327–335 (2006)
11. Zagal, J.P., Tomuro, N.: The Aesthetics of Gameplay: A Lexical Approach. In: *Proceedings of the 14th International Academic Mindtrek Conference*, pp. 9–16 (2010)
12. Zagal, J.P., Tomuro, N., Shepitsen, A.: Natural Language Processing for Games Studies Research. *Journal of Simulation & Gaming (S&G), Special Issue on Games Research Methods* 43(3), 353–370 (2011)
13. Zhan, T.-J., Li, C.-H.: Semantic Dependent Word Pairs Generative Model for Fine-Grained Product Feature Mining. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) *PAKDD 2011, Part I. LNCS*, vol. 6634, pp. 460–475. Springer, Heidelberg (2011)
14. Kayali, F., Purgathofer, P.: Two Halves of Play - Simulation versus Abstraction and Transformation in Sports Videogames Design. *Journal for Computer Game Culture, Eludamos* 2(1), 105–127 (2008)

Automatic Phone Alignment

A Comparison between Speaker-Independent Models and Models Trained on the Corpus to Align

Sandrine Brognaux^{1,2}, Sophie Roekhaut², Thomas Drugman³, and Richard Beaufort⁴

¹ ICTEAM - Université catholique de Louvain, Belgium

² CENTAL - Université catholique de Louvain, Belgium

³ TCTS Lab - Université de Mons, Belgium

⁴ Nuance Communications, Inc., Belgium*

sandrine.brognaux@uclouvain.be, sophie.roekhaut@uclouvain.be,
thomas.drugman@umons.ac.be, richard.beaufort@nuance.com

Abstract. Several automatic phonetic alignment tools have been proposed in the literature. They generally use speaker-independent acoustic models of the language to align new corpora. The problem is that the range of provided models is limited. It does not cover all languages and speaking styles (spontaneous, expressive, etc.). This study investigates the possibility of directly training the statistical model on the corpus to align. The main advantage is that it is applicable to any language and speaking style. Moreover, comparisons indicate that it provides as good or better results than using speaker-independent models of the language. It shows that about 2 % are gained, with a 20 ms threshold, by using our method. Experiments were carried out on neutral and expressive corpora in French and English. The study also points out that even a small neutral corpus of a few minutes can be exploited to train a model that will provide high-quality alignment.

Keywords: Alignment, Phonetics, HMM, Corpora, Annotation.

1 Introduction

Large speech corpora are required both in linguistic research and speech technologies. A characteristic of these corpora is that the sound cannot be studied alone. Most of the time, an orthographic and a phonetic transcription of the audio files are needed. The phonemes, in particular, should be synchronized with the sound. Indeed, the analysis of intonation, pronunciation, etc. requires to know the precise position of the phonetic temporal boundaries. Similarly, unit-selection and HMM-based speech synthesis techniques rely on the segmentation of the sound in phones or diphones. The quality of the generated voice strongly depends on the alignment accuracy. The phonetic alignment, also called forced alignment, can be done manually. However, this process has two serious drawbacks. First, it is time-consuming: from 130 [1] to 800 times real-time [2]. For large corpora of several hours, as used for speech synthesis or speech recognition,

* The study was carried out while Richard Beaufort was still working at the CENTAL (Université catholique de Louvain, Belgium).

the resulting manual annotation time would become prohibitive, which is economically impracticable. Secondly, a language expert is required for the task. The alignment process is not trivial and needs to be done as consistently as possible. This is even more problematic if different human annotators are working on a same corpus.

To overcome these problems, automatic alignment tools such as EasyAlign [3], SPPAS [4] or P2FA [5] have been developed. They allow the alignment to be both consistent and reproducible at a very low cost. Most of these tools rely on the acoustic modelling of the language with Hidden Markov Models (HMM). During the training, the acoustic model of each phoneme or group of phonemes of the language is built. During the alignment phase, these models are used to align an audio file with its phonetic transcription. The process is very similar to speech recognition techniques except that the phonetic transcription is known.

Generally, the acoustic models are trained on large corpora with several speakers. They account for an overall realization of the language which is not specific to one speaker or speaking style. Most automatic alignment tools provide the user with such speaker-independent models which can be used to align new corpora. This method has several disadvantages. First, the number of languages covered by the provided models is limited. Therefore, some corpora cannot be aligned. Secondly, the performance of the model strongly depends on the agreement between the training corpus and the corpus to align. If they are too different, the alignment quality may be low.

A way to alleviate these two issues is to train the model directly on the corpus to align. It offers the advantage of applying to any language and any speaking style. Besides, training the models on the speaker to align was proven to be highly profitable in speech recognition [6].

The aim of this paper is to investigate the quality of the alignment produced with a model trained on the corpus to align. This is done in comparison with the use of available speaker-independent models provided by recent alignment tools. The paper is organized as follows. Section 2 proposes an overview of the state of the art. Section 3 provides a detailed description of our method based on a training on the target corpus to align. The experimental protocol designed to evaluate our results is stated in Section 4. Results of our experiments are then shown in Section 5, providing an assessment of the proposed approach as well as a comparative evaluation with state-of-the-art techniques based on speaker-independent models. Finally Section 6 concludes and discusses further works.

2 State of the Art

HMM-based phonetic alignment has been pointed at as the most reliable technique for automatic phonetic alignment [7, 8]. It relies on speech recognition paradigms. Most existing alignment tools and studies [1, 3-5, 9] are based on the HTK toolkit [10] or similar toolkits like Julius [11]. HTK offers an implementation of HMM and methods for speech recognition and forced alignment. Both the training and the alignment developed for the experiments in this paper make use of HTK.

With such toolkits, the number of models to train can generally be defined by the user. Usually, each phoneme is linked to one model, called a monophone model (Table 1(1)). Three to five states represent the different stages of its realization: the transition

and the stabilization phases. However, the models can also be associated with phonemes in context, regarding the phonemes on the left and on the right (Table 1 (2)). They are called triphones and allow modelling the coarticulation phase. Their use, however, can be problematic: a lot of data is required to offer a good representation of each triphone. Besides, the augmentation in the number of models also increases the processing time. A solution is the use of tied-state triphones: the phonetic context of each phone is no longer modelled in terms of phonemes but in terms of classes (Table 1 (3)). Classes are generally articulatory characteristics that should be defined beforehand.

Beside the phoneme models, two specific models can be added (see Fig. 1). A silence model ('sil') represents silent pauses. Conversely to other phoneme models, it allows a direct transition from the second to the fourth state and a backward skip from the fourth to the second. Silences can be indicated in the phonetic transcription. The second specific model is a short-pause model ('sp'), which is a one-state model. Its emitting state is tied to the centre state of the silence model. 'sp' models are automatically inserted between the words. It allows to detect a silence that was not mentioned in the phonetic transcription.

Most research assessing the performance of HMM-based alignment use speaker-independent models [3, 4, 12, 13]. [13] offers an insightful comparison of the alignment rates when using various available speaker-independent models. The existing alignment tools (EasyAlign, SPPAS, P2FA, etc.) also provide the user with speaker-independent models, for several languages. These models can be used to align new corpora. The user has no access to the training stage and cannot train new models.

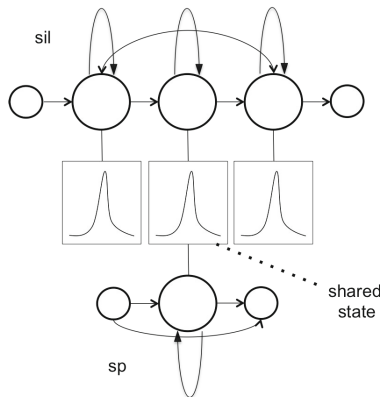


Fig. 1. Silence model and short-pause model

Table 1. Different model configurations

Models	Examples
(1) Monophones	[a]; [u]
(2) Triphones	[b-a+f]; [p-a+v]; [j-u+z]; [w-u+Z]
(3) Tied-State Triphones	[occlusive - a + fricative]; [semi-vowel-u+fricative]

A first drawback of such methods is that the number of available language models is limited. This means that many languages are not modelled and that the corresponding corpora cannot be aligned. The languages covered by some of the most widely-used tools (which will be presented in Section 4 and evaluated in Section 5.3) are shown in Table 2. It is worth noting that widespread languages, like Russian or German, are not covered by these tools.

A second flaw regards the quality of the provided models. They should be generic enough to produce high-quality alignment of different speech varieties or various speaking styles: neutral speech, spontaneous speech, expressive speech, etc. However, the model is strongly related to the corpus used for the training. Besides, if some phonemes were rare or misrepresented in the training corpus, these phonemes will be prone to alignment errors.

Table 2. Languages covered by various existing tools for automatic phonetic alignment

Tool	Language
EasyAlign	French, Spanish, Portuguese, Taiwan Min
SPPAS	French, English, Italian, Chinese
P2FA	American English

To alleviate these problems, we propose to train the model directly on the corpus to align. Few studies have evaluated the alignment quality obtained with such models. The performance of speaker-dependent models was analyzed in [1]. In this study, however, the model is trained on *aligned data* of one speaker and used to align some *other part* of a corpus of the same speaker. This improves the quality of the alignment because of a better agreement between the model and the corpus to align. However, it requires some part of the corpus to be manually-aligned. This is time-consuming, and hence costly.

The potential of training the model on the corpus to align was evaluated in [14] and [8]. It showed promising results, notably for its use for under-resourced languages [8]. However, it was not compared with the results obtained when aligning the same corpus with existing speaker-independent models. In [14], it was only tested on a corpus of 100 utterances and no claim was made about the minimum size required for the corpus to train a model. Results in [8] on African languages suggest that a small corpus of about 20 sentences would be enough. This remains to be proved on other Indo-European languages.

3 Our Method: Train and Align

Our method works as follows. In a first stage, the entire (unaligned) corpus to align is used to train a new language model. Acoustic parameters are extracted from the sound files and modelled¹. The phonemic models are five-state monophones. It implements

¹ The acoustic parameters are 12 Mel Frequency Cepstral Coefficients (MFCC) and their first and second derivatives.

both silence and short-pause models. In the second stage, these models are used to align the training corpus itself. For that matter, it makes a specific use of HTK methods for both the training and the alignment. The advantage of the method is that it can apply to any language or speaking style, as no pre-existing model is needed. Another benefit is that the training parameters can be modified. This method will be referred to as Train&Align-mono (**T&A-mono**) in the remainder of this paper.

The method is proposed with monophones but also with triphones (**T&A-tri**) and tied-state triphones (**T&A-tied**). In this latter approach, the phonetic context is defined in terms of classes. The list of the characteristics exploited in the method is shown in Table 3.

Table 3. Classes used to determine the context with tied-state triphones

Classes	Values
Type	Vowel/Consonant/Semi-vowel
Place of articulation	Bilabial/Labiodental/Alveolar/Palatal/...
Manner of articulation	Plosive/Fricative/Liquid/...
Voicing	Voiced/Unvoiced

Considering the phonetic context is an advantage in our method. Indeed, the use of pre-existing speaker-independent models makes it harder to use triphones.

In pre-existing models, all the triphones of the language should be present. If the corpus to align contains new triphones, the alignment process fails. However, the phonetic context coverage of the training corpus usually differs from the coverage of the target corpus, even if the training corpus is rather large. Obviously, this problem does not arise when the model is trained on the corpus to align. For pre-existing models, a particularly large corpus would be required to model every triphone. A solution might be to assign average values to non-existing triphones. However, that could harm the quality of the model and hence, the alignment.

4 Experimental Protocol

For the experimentation, Train&Align is used to align two corpora :

1. A neutral French-speaking corpus used in the LiONS unit-selection synthesis [15]. It consists of 510 speech files that are phonetized and manually aligned. The total duration is around 110 minutes.
2. The Woggle corpus [16], a corpus of American English. It contains expressive speech related to five emotions (angry, sad, happy, fear and neutral) uttered by five female speakers. It consists of 1,068 files for a total duration of 51 minutes. It was phonetized and manually-aligned by the first author of this paper. Its particularity is its high degree of variability.

The automatic alignment is evaluated in comparison with the manual alignment. The performance is measured as the percentage of boundaries that are similar in both alignments, with a certain tolerance threshold. In other words, accuracy metrics used in this work consider the proportion of alignment boundaries for which the timing error is lower than a threshold varying from 10 to 40 ms.

To allow an insightful interpretation of the performance, a few benchmarks should be considered. Large discrepancies are noticed between human-made alignments. Usually, 20 ms constitutes a limit above which the agreement rate is fairly high. Using this 20 ms threshold, [3] obtains agreement rates of about 81 % and 79 % for the alignment of a French and of an English corpus, respectively. On an Italian corpus, [17] find rates between 88 % and 95 %. It is also insightful to know which rate is sufficient for a speech corpus to be used for speech synthesis. In [7], it is shown that unit-selection based synthetic speech produced from a corpus aligned with a 92% rate with a 20 ms threshold was perceived as nearly as good as speech based on a manually-aligned corpus.

In a further experiment, Train&Align is compared to the use of existing speaker-independent models. The models used for the comparison come from VoxForge [18] and from recent alignment tools (EasyAlign [3], SPPAS 1.4 [4] and P2FA [5]).

1. **EasyAlign** provides a model for French but not for English. Its French model was trained on “30 minutes of unaligned multi-speaker speech for which a verified phonetic transcription was provided” [3]. The model consists of monophones.
2. **VoxForge** only provides a model for English. We used the latest version (June 15, 2012). It was trained on nearly 100 hours of read speech that were automatically phonetically transcribed but not aligned. The model consists of tied-state triphones.
3. **SPPAS** provides models for both French and English. SPPAS French model was trained on 8 hours of phonetically transcribed but not aligned speech from the CID and the AixOxCorpus. CID contains conversational speech while AixOxCorpus is made of read speech. SPPAS English model is the model of July 2011 provided by VoxForge. It contains about 85 hours of multi-speaker read speech. The corpus was automatically phonetically transcribed but not aligned. The models consists of triphones.
4. **P2FA** only provides an English model. It was trained on 25.5 hours of word-aligned speech from the Scotus corpus. This corpus consists of oral arguments from the Supreme Court of the United States. The model is made of monophones.

In Section 5.3, the SPPAS and EasyAlign tools were used to align the corpus as the end user would have done. For VoxForge and P2FA, which do not provide a user-friendly graphical interface, the models were used with HTK. The models were all provided with the correct phonetic transcription.

5 Experiments

Experiments are divided into three evaluations. First, we evaluate in Section 5.1 the performance of Train&Align on the French and the English corpus. Secondly, the minimum size of the target corpus to use is investigated in Section 5.2. Finally, Section 5.3 provides a comparative evaluation between Train&Align, and the five state-of-the-art speaker-independent models presented in Section 4.

5.1 Assessment of Train and Align

The three versions of Train&Align (mono,tri and tied) were applied on the French and on the English corpus. The alignment rates are shown in Table 4.

Table 4. Alignment accuracy of the French-speaking corpus and the English-speaking corpus with Train&Align-mono, -tri, and -tied

	Correct <10 ms	Correct <20 ms	Correct <30 ms	Correct <40 ms
French-speaking corpus				
T&A-mono	58.25 %	82.56 %	91.75 %	95.91 %
T&A-tri	60.74 %	84.23 %	91.9 %	96.27 %
T&A-tied	61.58 %	84.59 %	92.22 %	96.43 %
English-speaking corpus				
T&A-mono	42.84 %	63.22 %	77.97 %	86.8 %
T&A-tri	42.26 %	62.8 %	78.43 %	87.92 %
T&A-tied	42.44 %	62.84 %	78.6 %	87.99 %

The results on the French-speaking corpus exceed 80% for a 20 ms threshold. It is rather close to the inter-annotator agreement rates reported in [3]. However, it yields some major errors (>30 ms tolerance) which should be manually corrected. We can assume that only a quick manual check should be enough to produce high-quality alignment. This would largely reduce the required processing time.

For the English-speaking corpus, the correct alignment rates are significantly lower. This is due to the high variability of the corpus which contains several speakers and emotions. Section 5.3 examines whether low results are also found when the alignment is performed with speaker-independent models of English.

The results indicate that considering a larger phonetic context helps in modelling the language. For both corpora, an increase in the alignment rates with a threshold of 30 ms or more is observed. For the neutral French-speaking corpus, the use of triphones should clearly be recommended as it improves the overall quality of the alignment. For the English-speaking corpus, however, the alignment rates for smaller tolerance thresholds decrease. This could be due to the high variability of the corpus. In that respect, the phonetic context might not be the most relevant feature to take into account. The acted emotion or the position of the emphatic stresses could play a more significant role in the acoustic variation.

The results on the French-speaking corpus tend to be in line with [12]. They point out that the use of context-dependent models like triphones improves the alignment for small tolerance thresholds. In Table 4, the increase for 10 ms is clearly higher than for 40 ms, which indicates a clear increase in the precision of the alignment. Contrary to their study, we do not notice any degradation of the model with a tolerance of 20 ms. This might be due to the enormous size of their corpus, consisting of 1,037 speakers. The corpus may be big enough to ensure a very precise modelling of the phonemes that is partially damaged with the use of the phonetic context.

It is worth wondering whether these rather high alignment rates for the French-speaking corpus might be due to the size of the corpus to align. More than 100 minutes of speech are used to train the models. This provides a fair amount of occurrences for each phoneme. That question is now addressed in Section 5.2

5.2 Influence of the Size of the Corpus

The corpora that need to be aligned can be of a rather small size. This section investigates the minimal size of the corpus so as to build a high-quality model. This was studied on both test corpora. The total size of the corpus was gradually decreased and the alignment performance with Train&Align was assessed. The results for the French-speaking corpus are displayed in Fig. 2(1). They show that the quality remains rather stable up to a two-minute corpus, beyond which the alignment performance rapidly degrades. This short duration is due to the low variability of the corpus consisting of read neutral speech. A similar test on the English-speaking corpus displayed a sooner decrease, between 30 and 15 minutes (see Fig. 2(2)). On the whole, a few minutes of neutral speech seem to be enough to train and align a new corpus. This confirms the findings of [8] on African languages.

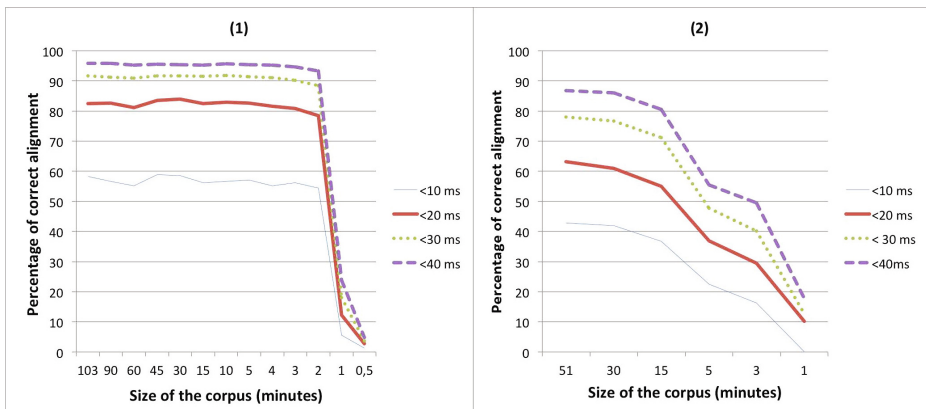


Fig. 2. Alignment rates with Train&Align-mono as a function of the size of the French-speaking corpus (1) and of the English-speaking corpus (2)

5.3 Comparison with Speaker-Independent Models

Train&Align is now compared to the five speaker-independent models presented in Section 4. It should be noted that :

1. Due to the conditions of distribution based on a GPL license, SPPAS uses Julius [11] and not HTK [10] to align the corpus. A disadvantage is that silences and short-pauses are skipped during the alignment stage. Silences are then processed separately, with the inter-pausal units (IPUs) segmentation tool. An orthographic

transcription must be provided to the tool with a specific label for silences. Those silences are detected on the basis of the signal only, in a phase that is independent from the alignment. The segments between the silences are aligned separately, with their supposedly corresponding transcription. However, silences are sometimes erroneously assigned to the signal. This penalizes the quality of the alignment as the system tries to align a signal with a transcription that does not correspond to it. To avoid such errors, only sentences for which the position of the silences was correctly detected were kept for the evaluation with SPPAS. It does not mean that the detected length of the silences was correct, but only that they were found at the right position. Both P2FA and EasyAlign use HTK and provide a silence model. A silence model is also implemented in Train&Align.

2. P2FA model depends on the lexical stress level of the phoneme. Three levels are considered: no stress, primary and secondary stress. Each vocalic phoneme is associated with three models. To exploit the full capacity of the system, all the phonetic transcripts were stress-annotated when aligning with P2FA model.

All the alignment tools used for the comparison do not provide models for both English and French. Table 5 shows the alignment performance on the French-speaking corpus with SPPAS model, EasyAlign model and Train&Align. Interestingly, Train&Align is observed to clearly outperform SPPAS and EasyAlign models across all measures. The gain compared to SPPAS goes up to nearly 15% with a 20 ms tolerance threshold. Besides, we know from Section 5.2 that the quality of the alignment remains stable up to a 2-minute corpus. It is striking to notice that training on 2 minutes of speech specific to the corpus to align provides better results than the use of a model trained on 8 hours of multi-speaker speech.

Table 5. Alignment accuracy of the French-speaking corpus with various models

Model	Correct <10 ms	Correct <20 ms	Correct <30 ms	Correct <40 ms
SPPAS	43.78 %	67.68 %	79.7 %	87.44 %
EasyAlign	54.18 %	80.7 %	90.27 %	94.28 %
T&A-mono	58.25 %	82.56 %	91.75 %	95.91 %
T&A-tied	61.58 %	84.59 %	92.22 %	96.43 %

Table 6 shows the alignment performance on the English-speaking corpus with SPPAS model, VoxForge model, P2FA model and Train&Align. A first observation that can be highlighted is that VoxForge significantly outperforms SPPAS that uses an earlier VoxForge model. This can be explained by the fact that a silence model is included in VoxForge model and processed by HTK. Errors made by SPPAS IPU segmentation are thus cancelled. A problem of VoxForge is that the model consists of triphones. The resulting flaw is that all triphones are not modelled and that some files cannot be aligned. Only 400 speech files out of the 1,068 files could be aligned. This problem was solved by SPPAS by adding unobserved triphones.

The improvement of the alignment quality with Train&Align compared to SPPAS and VoxForge is, here again, rather clear. The gain compared to VoxForge goes up to

nearly 15 % with a 20 ms tolerance threshold. However, it turns out that P2FA gives the best results, in particular for thresholds lower than 30 ms. This is probably due to the corpus used for the training, *i.e.* several hours of word-aligned speech. This is bound to improve the quality of the model. P2FA also takes different levels of stresses into account. This might improve the alignment as expressive speech displays more emphatic stresses. These stresses usually fall on the same position as primary stresses. It is again striking to notice that training on the (unaligned) corpus to align produces results that are comparable or slightly inferior to those provided by a model trained on more than 25 hours of word-aligned speech. The overall low alignment rate is clearly due to the high acoustic variability of the Woggle corpus.

On the whole, it is worth noting that Train&Align offers nearly as good or even better alignment of the corpus than existing tools used for comparison. This shows evidence that the alignment does not need to rely on existing speaker-independent models. This means that unseen languages or speaking styles could be automatically aligned.

Table 6. Alignment accuracy of the English-speaking corpus with various models

Model	Correct <10 ms	Correct <20 ms	Correct <30 ms	Correct <40 ms
SPPAS	11.04 %	26.25 %	49.39 %	70.6 %
VoxForge	23.78 %	48.56 %	70.85 %	84.82 %
T&A-mono	42.84 %	63.22 %	77.97 %	86.8 %
T&A-tied	42.44 %	62.84 %	78.6 %	87.99 %
P2FA	44.92 %	68.11 %	79.78 %	86.35 %

6 Conclusion

To align speech sound files with their phonetic transcription, HMM-based alignment methods have been developed. For the alignment of new corpora, pre-existent speaker-independent models, as provided by EasyAlign, SPPAS or P2FA can be used. However these models are only available for a very limited number of languages. Furthermore, they may produce low-quality alignments when used to align a corpus that strongly differs from the corpus used for the training (neutral vs. expressive, read vs. spontaneous, etc.). A solution offered by this article is to use the target corpus, which needs to be aligned, to train the acoustic model.

Several experiments showed that using a model trained on the target corpus yields nearly as good or even better results than using available speaker-independent models of the language. These available models were those provided by recent alignment tools: EasyAlign, SPPAS and P2FA, and the VoxForge English model. The improvement of our method was observed for neutral and expressive speech, as well as on both French and English corpora. Improvements in the alignment quality of about 2 % can be observed with our method with monophones (with a threshold of 20 ms). This can be explained by the fact that the model better captures the specificity of the target corpus. On the English-speaking expressive corpus, only P2FA outperforms our method, by about 5 % for 20 ms but only 2 % for 30 ms. This is due to their training corpus that consists

of more than 25 hours of word-aligned speech. The minimum size of the corpus to use to obtain high-quality alignment was also investigated. On a neutral speech corpus, it was found that only 2 minutes were sufficient to train the model properly. However, expressive speech is more variable and around 15 minutes of speech are needed. On the whole, this study points out that even small-sized corpora can be aligned without the need for pre-existing models of the language. The advantage is that this implies that any corpus in any language could be aligned autonomously, without alignment quality loss. The method also allows modifying training parameters like the model configuration. It was shown that the use of triphones instead of monophones further increases the alignment rates by about 2 % for large neutral corpora.

Other modifications of the training, left unexplored in this study, might also be applied. If a portion of the corpus is manually aligned, it could be used to improve the quality of the model, with bootstrapping methods. Ongoing tests show very promising results, especially on corpora for which low initial alignment rates were obtained, e.g. on the expressive English-speaking corpus.

If the target corpus includes several speakers or speaking styles, adaptation methods could also be applied. The models would be trained on the entire corpus and then adapted to each speaker or speaking style to align that specific part of the corpus. Obviously, these adaptation techniques could also be applied to the speaker-independent models offered by SPPAS, EasyAlign, etc. to improve the agreement with the corpus to align. This, however, requires the use of existing models that are not available for every language. Conversely, the objective of this study was to show that a corpus could be aligned autonomously, without damaging the quality of the alignment.

The results shown in this paper should be further confirmed by tests on other languages and speaking styles. A study is in progress on the alignment of Kirundi, an African language, and on a French-speaking corpus with different phonostyles (radio, sports, etc.). As previously mentioned, most user-friendly automatic alignment tools (SPPAS, EasyAlign, etc.) do not grant access to the training phase: it is impossible for the user to train a new model on the corpus to align. The tool we developed allows improving the results by training new models. It also offers a solution for languages for which no model is provided. This tool should be made available to the research community shortly.

Acknowledgements. Sandrine Brognaux is supported by the “Fonds National de la Recherche Scientifique” (FNRS). The authors would also like to thank Brigitte Bigi and Jean-Philippe Goldman for their help when using their tools and for their enthusiasm regarding this study.

References

1. Kawai, H., Toda, T.: An evaluation of automatic phone segmentation for concatenative speech synthesis. In: Proc. of ICASSP 2004, Montreal, Canada, pp. 677–680 (2004)
2. Schiel, F., Draxler, C.: The production of speech corpora. Technical report, Bavarian Archive for Speech Signals (2003)
3. Goldman, J.P.: Easyalign: an automatic phonetic alignment tool under Praat. In: Proc. of Interspeech 2011, pp. 3233–3236 (2011)

4. Bigi, B., Hirst, D.: Speech phonetization alignment and syllabification (SPPAS): a tool for the automatic analysis of speech prosody. In: Proc. of Speech Prosody 2012 (2012)
5. Yuan, J., Liberman, M.: Speaker identification on the SCOTUS corpus. In: Proc. of Acoustics 2008, pp. 5687–5690 (2008)
6. Leggetter, C., Woodland, P.: Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language* 9(2), 171–185 (1995)
7. Adell, J., Bonafonte, A., Gomez, J.A., Castro, M.J.: Comparative study of automatic phone segmentation methods for TTS. In: Proc. of ICASSP 2005, pp. 309–312 (2005)
8. van Niekerk, D., Barnard, E.: Phonetic alignment for speech synthesis in under-resourced languages. In: Proc. of Interspeech 2009, Brighton, pp. 880–883 (2009)
9. Cangemi, F., Cutugno, F., Ludusan, B., Seppi, D., Van Compernelle, D.: Automatic speech segmentation for italian (ASSI): Tools, models, evaluation and applications. In: Proc. of AISV, Lecce, Italy, pp. 337–344 (2011)
10. Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P.: *The HTK Book (for HTK Version 3)*. Cambridge University (1995)
11. Lee, A., Kawahara, T., Shikano, K.: Julius — an open source real-time large vocabulary recognition engine. In: Proc. of Eurospeech 2001, pp. 1691–1694 (2001)
12. Toledano, D., Gómez, L.: HMMs for automatic phonetic segmentation. In: Proc. of LREC (2002)
13. Chen, L., Liu, Y., Harper, M., Maia, E., McRoy, S.: Evaluating factors impacting the accuracy of forced alignments in a multimodal corpus. In: Proc. of LREC 2004, pp. 759–762 (2004)
14. Ljolje, A., Hirschberg, J., van Santen, J.: Automatic speech segmentation for concatenative inventory selection. In: Second ESCA/IEEE Workshop on Speech Synthesis, pp. 93–96 (1994)
15. Colotte, V., Beaufort, R.: Linguistic features weighting for a text-to-speech system without prosody model. In: Proc. of Interspeech 2005, pp. 2549–2552 (2005)
16. Dellaert, F., Polzin, T., Waibel, A.: Recognizing emotion in speech. In: Proc. of ICSLP, pp. 1970–1973 (1996)
17. Cosi, P., Falavigna, D., Omologo, M.: A preliminary statistical evaluation of manual and automatic segmentation discrepancies. In: Proc. of Eurospeech 1991, pp. 693–696 (1991)
18. MacLean, K.: VoxForge (2006-2012), <http://www.voxforge.org>

A Story Generation System Based on Propp Theory: As a Mechanism in an Integrated Narrative Generation System

Shohei Imabuchi and Takashi Ogata

Iwate Prefectural University, 152-52, Sugo, Takizawa, Iwate, Japan
g231k005@s.iwate-pu.ac.jp, t-ogata@iwate-pu.ac.jp

Abstract. We show the overall picture of Propp theory and the current state of system development. Specifically, we propose a story generation mechanism based on a Propp-based story grammar and propose an entertainment system KOSERUBE as the application. We aim at the integration of Propp theory into a story generation mechanism, another integration of Propp-based story generation system into a framework of integrated narrative generation system, and the pursuit of new types of contents with narrative generation functions.

Keywords: Narrative/story generation system, Propp, story grammar, story.

1 Introduction

This paper proposes a story generation system that unified with an integrated narrative generation system architecture [1] which is based on an interdisciplinary approach to narrative generation or “expanded literary theory” [2]. The integrated system divides a narrative generation process into conceptual structure generation part and surface expression part for language, visual media, and music. For the viewpoint of narrative characteristics, the former contains both story generation part and discourse one. This paper is related to the aspect of story. Events to be generated by the narrative generation system are organized into two kinds of structures: story and discourse. The organization of these structures is executed by micro level techniques to deal with elements such as partial relations among events and macro level ones to process more entire narrative structures. The proposed system which is a story generation system based on Propp’s narratological theory [3] is especially corresponding to the second level. The proposed system mainly generates narrative macro story structures by using a story grammar based on the Propp theory. However, at the same time, Propp theory and the story grammar also have micro level discourse relations such as “pair of functions”. Therefore, although the proposed system mainly provides a macro level knowledge to construct narrative overall structures, it also contains micro level knowledge to make the subtle narrative points.

V. Propp, who was a Russian folklorist, collected Russian folktales. His theory is one of the most influential origin and basis of structuralism, contemporary literary theories, and especially narratology which tries to pursue the forms and functions in

narratives. He considered folktales as a symbol of people's collective mentality and investigated its common structure and cultural characteristics. This paper combines the narrative analysis by Propp with a story generation system. We especially define a story grammar which is a core part for story generation based on the theory. It has a form of hierarchical generative grammar to generate stories by next three methods: top-down processing from the higher level to the lower one in the hierarchy, bottom-up processing from the lower level to the higher one, and the hybrid processing. Major differences from our previous research based on Propp [4] are to intend to integrate organically this mechanism into our narrative generation system architecture, which is a framework of narrative generation containing diverse components, and integrate narrative knowledge such as agents and objects in Propp's narrative world into a common conceptual dictionary. At the same time, these are original characteristics of this study to similar studies. Moreover, we aim to utilize a variety of mechanisms in the Propp theory as an organized literary theory comprehensively. The theory contains mechanisms for consistent story generation, complicated story generation, plausible scene development, and so on. Although only the aspect of "function" as shown later is emphasized, the theory has a variety of aspects. For a contribution to literary area, computational approach such as story generation system provides a method to treat them as one framework organically.

2 Related Studies

Propp theory has been introduced into information area such as AI and cognitive science, furthermore applied to story generation systems [5]. Schema theory in cognitive science means the framework of knowledge which human preliminarily holds and controls the cognition, and is the headstream of knowledge representation in AI. Narrative and story were also interpreted as the schema for human cognition in the world. Propp theory has an influence on story schema and story grammar.

Next, narratology is a research area which sights on the aspect of technique and form more than the content to be narrated and a kind of revival of rhetorical methods in literary theories. Propp theory has been accepted as a method which gives greater importance to the aspect of structural order in stories. Propp collected plenty of examples of Russian folktales to discover the common features through the cognition of structural invariable ("function" to be described). The main application to literary area was to adapt this aspect of Propp theory to various genres of narrative texts to aim at the theory's deepening and expanding. However, we recognize the theory is a research of the deconstruction of elements in a story. It contains or indicates many rhetorical methods other than "function" which forms the core of the theory. Through a constructive method of story/narrative generation, we extremely aim to reconstruct it as techniques for story/narrative generation by anatomizing the theoretical elements [4].

As a study of story generation system, KIIDS [5], which is a recent system on story generation applying Propp theory, utilizes the knowledge on "function" and actors' types as ontology to generate Russian folktale like plots. Fabulist system [6] applies

the Propp's "function" to manage the logical causal progression of a plot the characters' believability. Whereas, our framework or design of story generation system using the Propp theory is to aim at the more comprehensive application including "pairs of functions", techniques for combining some stories into one story or "the ways in which stories are combined" and so on [4]. Although only a part of the elements is implemented in a proposed system in this paper, we aim at developing a framework for integrating the theory by the deconstruction and reconstruction.

Finally, Turner [7] described that we cannot perform the task of story/narrative generation based on structural and formal knowledge like the Propp theory and the knowledge on content is required. He developed a story generation system MINSTREL for verification based on semantic knowledge fragments and case-based reasoning. This opinion is a true in a sense. However, in many cases, innovation in novels and arts is brought by concern with the structural or formal aspect and it also plays an important role for story/narrative generation. For example, narrative genres like detective story are considerably generated based on a structural or formal principle and it also contributes to compose a narrative efficiently and make a narrative longer. Both structural or formal knowledge and semantic one on content are required in narrative generation. In our integrated narrative generation framework, Propp-based story generation mechanism is mainly corresponding to the structural and formal aspect.

3 Toward Internal and External Integration of Propp Theory

The design of a story generation system based on Propp is divided into two aspects: the first is a comprehensive and organic integration of elements of the theory and the second is its integration with our integrated narrative generation system. We call the former "internal integration" and the latter "external integration". We describe the former here.

For the theory, the most important concept is "function", which means the action of an actor seen from the result. A function is corresponding to an abstract meaning for many concrete and actual actions. The structure of a story is described by a sequence of functions having basically a same order in the genre of Russian fairy tale. Propp proposed 31 kinds of functions¹ and one additional function (first "preliminary part") for defining the structure and described that the order in narratives is principally same. The theory gave a great influence to the later structuralism by insisting many variations in narratives as phenomena can be realized by the existence of a few structural narrative elements, which originates from the "functions".

¹ The 31 functions contain: (1) **Absentation**, (2) **Interdiction**, (3) **Violation**, (4) **Reconnaissance**, (5) **Delivery**, (6) **Trickery**, (7) **Complicity**, (8) **Villainy or lack**, (9) **Mediation**, (10) **Beginning counter-action**, (11) **Departure**, (12) **First "function" of donor**, (13) **Hero's reaction**, (14) **Provision or receipt of a magical agent**, (15) **Guidance**, (16) **Struggle**, (17) **Branding**, (18) **Victory**, (19) **Liquidation**, (20) **Return**, (21) **Pursuit**, (22) **Rescue**, (23) **Unrecognized arrival**, (24) **Unfounded claims**, (25) **Difficult task**, (26) **Solution**, (27) **Recognition**, (28) **Exposure**, (29) **Transfiguration**, (30) **Punishment**, and (31) **Wedding**.

However, Propp is originally a folklorist and the theory is supported by enormous actual folktales. Although the central part is the aspect of structuralism, the research as wholeness contains other various methods. Propp shows plural examples or techniques for actualizing “functions” in the comparatively lower level based on collected folktales. For instance, as a way for realizing the “function”, “Interdiction”, Propp shows “interdiction” and “order or a suggestion”. We call this level for realizing a “function” “sub-function”. Moreover, he describes that a specific “function” evokes another specific “function” and as the result both functions form one pair. This means a specific “function” has a strong tendency that it calls another “function” in a story generation process. Such functions are called “a pair of functions”². In addition, a specific “function” is assumed by a specific type of actor. The types of actors are “hero, villain, victim, helper, dispatcher, princess and false hero”. Each actor is abstracted as roles for realizing some “functions” and a specific role becomes the main agent of a group of “functions”.

We [4] regarded the Propp theory as an integration of literary techniques or methods from structural and macro level to micro level, and classified its methodological elements as shown in Fig. 1. Through previous relating studies ([4]), as described above, we saw the Propp theory as a kind of deconstruction of narrative elements and proposed next prototyping programs as its reconstruction: (1) a system using “functions”, the hierarchy, pairs of “functions” and seven types of actors, (2) a system using techniques for combining some stories into one story or ways in which stories are combined, and (3) a system emphasizing actors’ roles. However, we did not develop one integrated system in which the above elements are organically combined because of the difference of data structures and processing methods. Moreover, the above programs could not be incorporated in the integrated narrative generation system framework. Therefore, the next goals of us are first type of system integration within the Propp theory including the above elements especially and second type of system integration with a narrative generation system framework. In this paper, we deal with (1), (2), (3), (4), (11), and (17) in the elements as shown in Fig. 1 and combine the system with the narrative generation system. The former is chiefly done by defining a story grammar.

4 The Present State of the System Development

We propose a renewal of Propp-based system by [8]. The characteristics are a story grammar which is a systematic organization of the theory and conceptual dictionaries with two hierarchies for verb and noun concepts. The dictionaries have been developing to be commonly used in the integrated narrative generation system. This Propp system is designed to be unified with the whole system through the dictionaries. Fig. 2 is the configuration. A story is a sequential events binding hierarchically by narrative relations including causal relation, continuation relation, and so on.

² The pairs contain: **Interdiction/Violation, Reconnaissance/Delivery, Trickery/Complicity, Villainy or lack/Liquidation, First Function of donor/Hero’s reaction, Guidance/Return, Struggle/Victory, Pursuit/Rescue, Branding/Recognition, and Difficult task/Solution.**

(1) “Functions” and its chained rules, (2) A concrete way for actualizing the “function” (“Sub-functions”), (3) “function” pairs, (4) “Sub-function” pairs, (5) Auxiliary elements for the inter-connection of “functions”, (6) Motivation, (7) Ways in which new actors are introduced into the course of action, (8) Inversion of “functions”, (9) Position change of “functions”, (10) Chasm of “functions”, (11) The distribution of “functions” among dramatis personae, (12) Auxiliary elements in trebling, (13) Ambiguity of “function”, (14) Anabolism of “function”, (15) Attributes of dramatis personae, (16) Combination of several stories (moves), (17) The generation model of the folktale, (18) Restraint and freedom in a narrator, (19) Modification of the folktale.

Fig. 1. Elements in Propp Theory

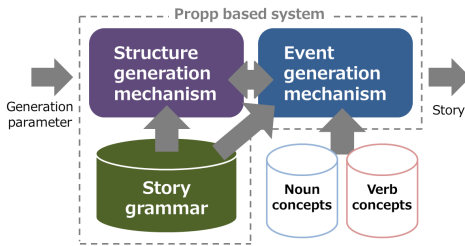


Fig. 2. The system configuration

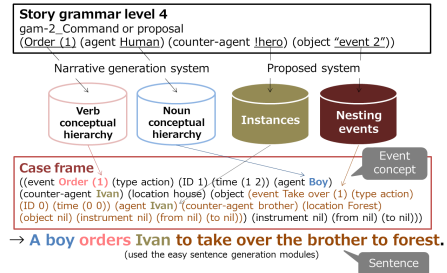


Fig. 4. An event concept generation using conceptual dictionaries

```
(setq *Propp-level1-list* '((Russian folktale (00_Preliminary part P-Problem P-Trial P-Solution))))
(setq *Propp-level2-list* '((Problem (Reserve portion Beginning)
  (Trial (OR (Reserve trial Battle and victory) (Reserve trial Task and solution))
  (Solution (Problem dissolution Arrival and trial End))
  (Reserve portion (OR (01_Absentation 02_Interdiction 03_Violation) ...
(setq *Propp-level3-list* '((00_Preliminary part (alp-1_Preliminary part)
  (01_Absentation (OR bet-1_Going out1 bet-2_Death bet-3_Going out2))
  (02_Interdiction (OR gam-1_Interdiction gam-2_Command or proposal) ...
(setq *Propp-level4-list* '((alp-1_Preliminary part ((Preliminary part (1))))
  (bet-1_Going out1 ((Go out (1) (agent Parents) (object Business))))
  (bet-2_Death ((Dead (1) (agent Parents)))) ...
(setq *Propp-function-pair-list* '((gam-1_Interdiction del-1_Violation)
  (gam-2_Command or proposal del-2_Command execution)
  (eps-1_Inquires1 zet-1_Instruction1) ...
```

Fig. 3. A part of story grammar

An event is a frame representation, which is constructed by a verb concept and case elements for noun concepts such as agent, object, location, and so on. The verb and noun concepts are associated with a set of conceptual dictionary containing verb concepts hierarchy and noun concepts one.

4.1 A Story Grammar from Propp Theory

We defined a story grammar by the restructuring or reorganization of the Propp theory. Fig. 3 shows a part of the story grammar. It is divided into next four levels of

a hierarchy. The level 1 is the highest level to determine the macro or overall structure of a story. The level 2 is the layer for grouping 31 “functions” into some parts. The level 3 generates 175 kinds of “sub-functions” from the “functions”. While a “function” means abstract and comprehensive definition of an actor’s action, a sub-function means concrete methods for actualizing a “function”. The story grammar also implements pairs of functions. Actually, 82 kinds of pairs are defined at the level of sub-functions. In the last level 4, each sub-function is derived to an action or a sequence of actions (221 kinds) in which each action is described in a case frame of event that has a verb concept and necessary noun concepts. A nested event having some verb concepts is also processed in a special mechanism. These semantic elements are linked to conceptual dictionaries for hierarchies of verb and noun concepts. In addition, an agent in a case frame under each “function” is corresponding to one of seven roles or agent types including hero, villain, victim, helper, dispatcher, donor, and false hero.

4.2 Combining with Conceptual Dictionaries

As explained above, the narrative generation system uses conceptual dictionaries which has two kinds of hierarchies for verb and noun concepts. The verb concept hierarchy has 4260 concepts and each concept defines a case frame representing necessary cases and constraints for values of the each case. Constraints mean the range in the noun concept hierarchy for the cases. On the other hand, the noun concept hierarchy has about 120000 noun concepts. Oishi et al. [9] provide the detailed explanation. The each event concept is represented by a set of case frame containing a verb concept and some noun concepts. In the story grammar, events are generated in the level 4. We show the generation process of an event concept using the conceptual dictionaries based on Fig. 4. First, the system searches for a verb concept described in the level 4 to get the case frame (For example, “Order (1)” for “Interdiction”). Next, the system inserts the case information described in the level 4 into this case frame. At this time, if there is a description (like !hero) enclosed by asterisks corresponding to the seven types of actors’ roles, the system assigns a name (like “Ivan”) to a variable. If there is a description enclosed by double quotes (like “human”), the system refers the noun concept hierarchy to get a lower level’s concept (like “boy”). The description of event contained in a case frame such as “event 1” means a nested event concept. The system inserts a new event concept into the case frame. Finally, a simple sentence generation program transforms the case frame to a sentence.

4.3 Generation Processes

The story grammar is opened to different types of operations such as top-down, bottom-up, and hybrid. In the all mechanisms, an events generation mechanism generates concrete events in the processing in the level 4. In addition, a simple sentence generation program transforms each event concept into a sentence principally. The top-down process is the standard method and simply expands the story grammar from the higher level to the lower level according to the user’s input information. When the

processing reaches to the lowest level, the system generates an event concept. In the bottom-up processing, the user inputs a case frame, which is corresponding to the lowest level in the story grammar, and the number of hierarchy to be rose and the system makes a structure by using the hierarchy and the pairs. We describe the hybrid method in detail.

This is a mixed method of the top-down mechanism explained above and bottom-up processing which goes up the structure of story grammar from a point in the lowest hierarchy in the story grammar. We show the flow of processing: (1) The user inputs the names of actors corresponding to seven types of roles, an event concept in the level 4, and the number of hierarchy to go up (from 1 to 5). If the number of hierarchy is 1, the processing goes up the layer of “sub-function”, and the layer of “Russian folktale” is corresponding to 5. In addition, we use “the number of hierarchy: 4” here. (2) The system collates this input event concept with elements at the level 4 in the story grammar to go up the hierarchy according to the number. In the above example, the processing goes up the sub-function, “A-1_Abduction”, the “function”, “08_Villainy”, “Beginning” and “P-Problem” to make the smallest structure of a story. If an event concept is commonly used for some sub-functions, some skeletons are severally generated. (3) For the bottom-up processing, when the system meets an element in a pair of sub-functions, the system automatically expands another element in the pair. And the system goes up to the directed hierarchy from the expanded sub-function to extend the story’s skeleton. In this example, “K-1_Seize1”, which is the pair of “A-1_Abduction”, is expanded and the processing goes up to “19_Liquidation”, “Problem dissolution” and “P-Solution” in the level 2 from the point. The above is corresponding to bottom-up processing to go up the story grammar’s hierarchy. (4) Based on the structure(s) made finally in the bottom-up processing, the system again goes down the hierarchy to complete the story’s structure. If the input information is from hierarchy: 1 to hierarchy: 4, a partial story is generated. If it is hierarchy: 5, a complete story containing all elements in the lower hierarchy of “Russian folktale” is generated. In the above processing, if top-down processing is not added, it means pure bottom-up processing. In the Propp theory, as all “functions” do not have to be used, the stop of processing at an incomplete level is not necessarily problem.

5 An Execution Example

Fig. 5 shows an example of input and output in the top-down processing. The parameter is 3. The content of system output are: (1) sentences by the simple sentence generation program, and (2) generated event concepts to be sent into sentence generation program. The length of story changes according to the user’s input. To generate longer stories, we will be able to use some techniques like “the ways in which stories are combined” [3][4], “repetition” in narrative discourse mechanism [10], and so on.

A problem is that there are parts in which a kind of jump in a story flow exists in the connection among events. It may be true that such jump is not necessarily a problem in narrative, especially in folktale narrative. However, there is the technique for inserting additional event or events into a point in a story to compliment among events. Propp also describes the additional events to connect among “functions” [3].

Input (Propp system):
(st-propp '(Ivan warrior@brave) (Snake snake@*reptilian[snake]*) (Princess princess@woman) (Magic_wand wand@*stick[wand]*) (King king@*crown[king]*) (Baba_yaga beldam@woman) (Melos man@man)) 'PROPP 'topdownB '3 'nil)

Output (Propp system):
(\$ロシア民話 (\$問題 (\$予備部分 (event 出かける1 (type action) (ID 1) (time (time1 time2)) (agent age%子供#1) (counter-agent nil) (location loc%巣箱#1) (object obj%用事#2) (instrument nil) (from nil) (to nil)) (event 乱暴する1 (type action) (ID 2) (time (time2 time3)) (agent age%蛇#1) (counter-agent age%勇士#1) (location loc%里#1) (object nil) (instrument nil) (from nil) (to nil)) (event 負傷する1 (type action) (ID 3) (time (time3 time4)) (agent age%勇士#1) (counter-agent nil) (location loc%里#1) (object nil) (instrument nil) (from nil) (to nil))) (\$発端 (event 欠如する1 (type action) (ID 4) (time (time4 time5)) (agent age%皇女#1) (counter-agent age%勇士#1) (location loc%塔#1) (object nil) (instrument nil) (from nil) (to nil)) (event 来る2 (type action) (ID 5) (time (time5 time6)) (agent age%勇士#1) (counter-agent nil) (location loc%王朝#1) (object nil) (instrument nil) (from loc%里#1) (to loc%王朝#1)) ...

Converted sentence (by sentence generation module in the narrative generation system):
A child goes out to an affair. A snake does violence to a warrior. The warrior is wounded. The warrior lacks a princess. The warrior comes to dynasty from a village. A king calls for help to the warrior. The king sends the warrior. The warrior prepares for an adventure. The warrior departs to the adventure. The warrior heads over to a government office. A wounded person asks the warrior to hold a memorial service. The warrior does not hold the memorial service for a dead man. An older woman advices on a stick to the warrior. The warrior uses the stick. The warrior flies. The warrior heads over to an enemy territory. (… omitted) A deceit by a man is exposed. The warrior puts on a running shirt. The warrior is promoted to executive. The king forgives the snake. The king forgives the man. The warrior gets a country.

Fig. 5. An example of generated stories (translated into English by hand)

6 An Application System and the Preliminary Evaluation

We intend to use the Propp-based mechanism as a module in the integrated narrative generation system and adapt to application systems. Actually, we have applied it to an experimental integrated narrative generation system and developed an application system in which it is the main method. In the integrated narrative generation system, a story and a discourse are respectively represented with a tree structure in which the lowest nodes are events and the other nodes are relations. A Propp-based story structure can be also inserted into the structure organically. This section describes the application.

The application system which we call it KOSERUBE is a system that generates automatically stories and discourses with characters, places and objects relating to Iwate prefecture in Japan, the sentences and music, and edits automatically visual objects relating to generated narratives. The user can operate and appreciate the process through the human interface. In addition, KOSERUBE means “let’s make (stories)” in Iwate’s dialect. Specifically, as the input information, the user selects a hero and a villain, the length of a story to be generated, and a narrator. The user can also read the explanation about characters and places in Iwate prefecture. In the automatic generation mode, a narrative, the sentences and music are generated and expressed on stage like display by the automatic editing of visual objects. Basically, each sentence of event in a generated narrative are represented using caption, reading aloud, characters’ pictures, background pictures, music in order. In addition, we prepare the following additional functions: a curtain for showing temporal changes, explanation and description about the main objects, the insertion of meta-description like “three days have passed”.

Especially, an important adding function relating to the Propp-based mechanism is a mechanism for customizing narrative patterns. In researches that aim at the generalization of the Propp theory, for example, structures of a kind of Japanese folktales can be described by using Propp’s “functions”. For example, “The Grateful Crane” has a series

of functions of “Lack - Difficult task - Solution - Liquidation - Absentation - Violation - Lack”. The story grammar can generate stories that have such structure by substituting this structure for the second hierarchy. The generated stories have this story line, but the more concrete events are different from the content of the folktale.

To investigate the effects, issues of the above system to be solved, we conducted (1) a questionnaire survey in 163 students of information area (having no knowledge on AI) and (2) the exhibition to ordinary persons at Yamaguchi Center for Arts and Media. As the former’s result, the average of interestingness of generated narratives was 2.3 for 1 to 4. We acquired two types of contradictory comments that generated narratives are sometimes rapidly and have no consistency, but in contrast they are very interesting as the surreal characteristic. The system needs the ability for creating consistent narratives and the mechanism for destroying such consistency intentionally. This problem will be improved by revising the sequence of “functions” and the pairs, introducing the methods for connecting two “functions” smoothly, adding characters’ motivations for the occurrence of an event, and so on. In the (2), many children have participated and we acquired unexpected reactions. Some children directly participated by standing on the front of the display and attempted repeatedly specified actions which they especially had an interest. Although the system has many issues to be improved technologically, we would like to consider the applications to comparatively simple domain like an elementary English learning system with a narrative generation mechanism.

7 Conclusion

We showed the overall picture of Propp theory and the current state of system development. Moreover, we have experimentally implemented an entertainment system, KOSERUBE. From a broad view point, we showed a direction toward entertainment contents of the narrative generation research. One of issues in the next step is to introduce comprehensively the elements of Propp theory into the system and use this system organically in the integrated narrative generation system.

References

1. Akimoto, T., Ogata, T.: A Consideration of the Elements for Narrative Generation and a Trial of Integrated Narrative Generation System. In: Proc. of the 7th International Conference on Natural Language Processing and Knowledge Engineering, pp. 369–377 (2011)
2. Ogata, T., Kanai, A.: An Introduction of Informatics of Narratology. Gakubunsha (2010) (in Japanese)
3. Propp, V.: Morphology of the Folktale. University of Texas Press (1968)
4. Ogata, T.: To the Rhetoric of Story from Propp. *Cognitive Studies* 14(4), 532–558 (2007) (in Japanese)
5. Peinado, F., Gervás, P.: Creativity Issues in Plot Generation. In: Workshop on Computational Creativity. Working Notes. 19th International Joint Conference on AI, pp. 45–52 (2005)
6. Riedl, M.O.: Incorporating Authorial Intent into Generative Narrative Systems. In: Proc. of the AAAI Spring Symposium on Intelligent Narrative Technologies II (2009)

7. Turner, S.R.: *The Creative Process: A Computer Model of Storytelling and Creativity*. Lawrence Erlbaum (1994)
8. Imabuchi, S., Ogata, T.: Story Generation System based on Propp Theory as a Mechanism in Narrative Generation System. In: 4th IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning, pp. 165–167 (2012)
9. Oishi, K., Ogata, T., Onodera, K.: Towards the Development of Conceptual Dictionary for Narrative Generation System. In: Proc. of the 7th International Conference on Natural Language Processing and Knowledge Engineering, pp. 351–358 (2011)
10. Akimoto, T., Ogata, T.: Computational Model of Narrative Discourse Theory and Reception Theory in Narratology and its Implementation. In: Conference Handbook of the 13th Annual International Conference of the Japanese Society for Language Sciences, pp. 155–156 (2011)

Automatic Utterance Generation by Keeping Track of the Conversation's Focus within the Utterance Window

Yusuke Nishio and Dongli Han

Department of Computer Science and System Analysis, College of Humanities and Sciences,
Nihon University, Tokyo, Japan
han@cssa.chs.nihon-u.ac.jp

Abstract. The insufficiency in methods for generating utterances still remains as a critical issue unsolved in the community of non-task-oriented conversation. Previous studies provide various strategies to enrich the methods for generating utterances, thus making the conversation systems or agents appear more interesting. However, none of them could escape from the fact that they all generate utterances depending mainly on some particular kinds of templates or augmented templates. We propose here in this paper a thorough modification to a preceding work to address this problem. Specifically, we first introduce a concept Utterance Window to strengthen the association between continuous utterances, and then employ a Two-starting-word Markov connection to cope with the ease of losing focus of the current utterance. In addition, we try to keep track of user's interests and reflecting them in the process of topic-word extraction and utterance generation as well. The experimental results show the effectiveness of our method.

Keywords: Topic Word, Utterance Window, Utterance Generation, Conversation System, User's Interest.

1 Introduction

During the past decade, a number of non-task-oriented conversation systems have been developed, whereas the insufficiency in methods for generating utterances still remains as a critical issue unsolved. Non-task-oriented conversations pay more attention to continuing the current conversation by any means rather than the rigorousness of the utterance's content in comparison with task-oriented ones. Researchers must try to make their computer-generated utterances in more various forms and more interesting to keep the users talking.

Here, we give a review on some previous efforts for utterance generation in non-task-oriented conversation systems. Fujimoto et al. make an analysis on the conceptual relations between an utterance and the topic contained, and present an approach to change topics based on conceptual relations in a free conversation system [1]. Saito et al. employ a method to enrich the content of the conversation by reorganizing the real world information that has been obtained ahead [2]. Higuchi et al. concentrate on modalities appearing in human's utterances, and try to incorporate them to the process

of utterance generation [3]. Yoshioka et al. guess the user's character and preference through the conversation and try to use the obtained information to improve the system performance [4]. Song et al. [5] and Han et al. [6] present a strategy to provide new topics for users in a free conversation system at the point the system "considers" that the user has lost interest in the current topic.

The above studies provide various strategies to enrich the methods for generating utterances, thus making the conversation systems or agents appear more interesting. However, none of them could escape from the fact that they all generate utterances depending mainly on some particular kinds of templates or augmented templates. In other words, the monotonicity of computer-generated utterances hasn't been addressed ideally, and the conversation agent still sounds like a machine rather than a natural human.

With this knowledge in mind, a study has been made by using Markov sequences in order to generate more flexible utterances [7]. In another research, Han et al. develop a free conversation system as shown in Figure 1 also employing Markov sequences but in a completely different manner [8]. In Figure 1, Planning Block determines the topic-word to be used in utterance generation, Generating Block generates the computer-utterance, and Encoding Block assigns human-like characters to the generated computer-utterance. This system obtains the n-gram data from Twitter, a virtual community where one can find a huge number of real utterances by humans, and tries to generate more natural utterances of greater diversity and flexibility.

Although the method in [8] has been proven quite effective in promoting the human-like qualities of utterances, two main problems have been observed simultaneously: weak association between continuous utterances, and ease of losing the focus of the current utterance. The former one is due to the nature of the method that each utterance is generated taking consideration of the latest user-utterance only. The latter one comes probably from the Markov connecting algorithm where a whole sentence is formed by starting from a single word in the center and spreading through the n-gram network both to the left and the right until it encounters the beginning signal or ending signal finally.

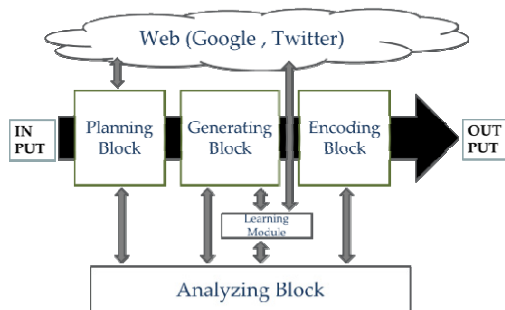


Fig. 1. System Model of A Preceding Work [8]

What we propose here in this paper is a thorough modification to the preceding work [8]. We add another two modules into the original framework: the Extracting Block, and the Multiple-word Generating Module to cope with the problems described above as shown in Figure 2. Specifically, we introduce the concept Utterance Window to strengthen the association between continuous utterances in the Extracting Block, and employ a Two-starting-word Markov connection to cope with the ease of losing the focus of the current utterance in the Multiple-word Generating Module. In addition, we try to keep track of user's interests and reflecting them in the process of topic-word extraction and utterance generation as well. We describe in detail the above ideas in Section 2, 3, 4 and then give some experimental results for verifying their effectiveness in Section 5.

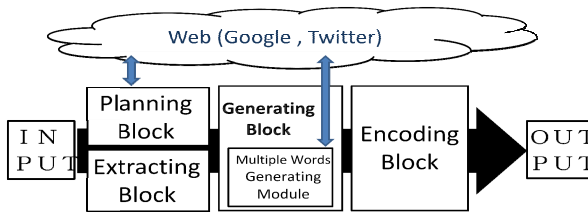


Fig. 2. Modification Diagram

2 Utterance Window

In the preceding work, each new utterance in a conversation system is generated simply according to the last user-utterance only [8]. We believe this is the reason why the preceding system performed frailly in holding continuous topic and keeping the conversation constant. In this paper we introduce a concept called Utterance Window to take a bunch of continuous user-utterances into consideration simultaneously before the system is going to decide topic words for the next computer-utterance.

When we try to extract topic words from a series of continuous user-utterances, to what extent should we go back from the current user-utterance? In other words, how many user-utterances should we consider as the scope from which to extract topic words for the future conversation? Then we refer to the work of Kurohashi et al. where a technique was developed to investigate the word's density within a certain range of text with weights assigned by Hanning window function [9], and come up with the idea of incorporating an Utterance Window into the framework. We believe that a window with appropriate size, i.e., the total number of user-utterances contained in it, will be more ideal for topic-word extraction than a single user-utterance which was employed previously and turned out to cause the weak association between continuous utterances. In our system, we keep retrieving topic words from moving Utterance Windows as Figure 3 shows. Here an Utterance Window contains n user-utterances. A larger n implies the employment of older user-utterances, which might yield more noises from the old part of the conversation. We discuss the ideal number for n through an experiment and describe the results later in Section 5.

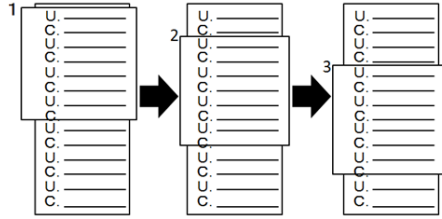


Fig. 3. Moving Utterance Window

3 Extraction of Topic Words

This section describes in detail the steps taken to extract topic words from the user's Utterance Window for the generation of the next computer utterance. Here, we first introduce two concepts: interest recognition factor and time decreasing factor, representing the user's current interests, and then describe a manner to assign weights to topic-word candidates within the current Utterance Window based on User's Interests. Finally, as we will be using topic-word pairs for further process, a method to determine the association degree between topic-word candidates is proposed with some discussions.

3.1 User's Interest

The User's interest could be represented by words occurring in the Utterance Window. Here, we introduce two concepts to help understand the relations held among time, word counts, and the user's interest. One concept is the interest recognition factor, and the other is the time decreasing factor. These two concepts were first introduced by Satake et al. to select contents from the interactive news [10]. The interest recognition factor pays more attention to words with higher counts in the user's utterance. On the other side, the time decreasing factor points out the importance of newly stated words in the user's utterance, i.e., words that didn't show up in the latest user's utterance might have less relation with the user's current interest comparing with those do.

3.2 Weight Assignment of Topic Word Candidates

With these concepts in mind, we define a measure below to calculate and assign weights to the j -th topic word candidate W_j as shown in Figure 4. Topic word candidates include nouns, verbs, and adjectives occurring in the current Utterance Window.

$$S_{ij} = \frac{l-i}{l} \times C_{ij}$$

$$T_j = \sum_{i=0}^{l-1} S_{ij}$$

Here, l indicates the length of the Utterance Window, i.e., the number of user-utterances contained in it, and i indicates the sequence number of each user-utterance in the Utterance Window. C_{ij} represents the counts of W_j in the i -th user-utterance, S_{ij} represents the score, i.e., the weight assigned to W_j for the i -th user-utterance, and T_j indicates the total score of W_j in the current Utterance Window.

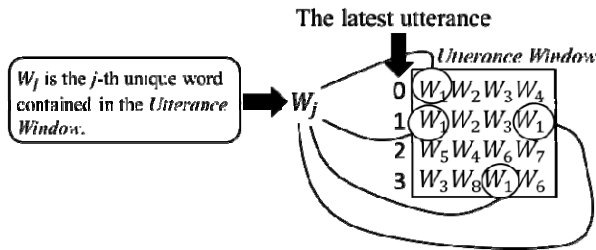


Fig. 4. An Illustrative Example for W_j ($j=1$)

3.3 Association between Topic Word Candidates

A major difference from the preceding work of the present algorithm as described in Section 1 lies in the point that we decide to use two starting words in later Markov connection process. We expect to improve the situation that a computer-utterance tends to lose focus in this manner.

Our purpose is to select two words as the starting points for later Markov connection process. A simply way to accomplish this could be just rank all the topic-word candidates by their weights and grab the first two. However, this doesn't guarantee at all a close association existing between the top two words. We need two highly-associated starting words rather than mutually unconcerned word pairs to generate a meaningful and semantically focused utterance sentence. The formula below comes out from the Mutual Information [11] and is defined here to measure the degree of association between two topic-word candidates W_a and W_b based on their counts within the Utterance Window.

$$SR(W_a, W_b) = \frac{\sum_{i=1}^{f(W_a)} \sum_{j=1}^{f(W_b)} \left(\frac{1}{|N_i(W_a) - N_j(W_b)| + 1} \right)}{f(W_a)f(W_b)}$$

Here, $f(W_a)$ and $f(W_b)$ indicate the counts of W_a and W_b in the current Utterance Window, $N_i(W_a)$ and $N_j(W_b)$ represent the sequence number of the user-utterance where the i -th W_a and the j -th W_b show up respectively.

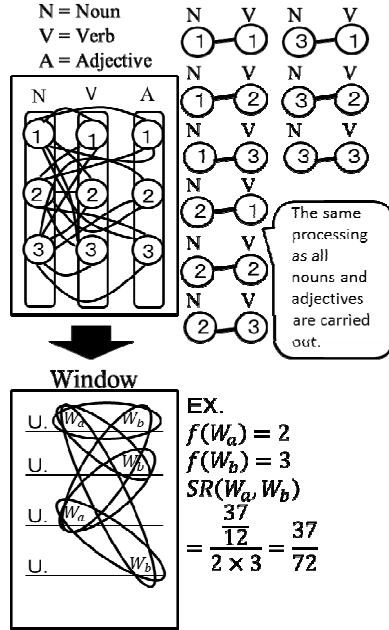


Fig. 5. Association Calculating Process and An Illustrative Example

The real system grabs the top three words for each of noun, verb, and adjective based on the computing results of the formulas defined in Section 3.2, and then calculates the degree of association for each noun-verb pair and noun-adjective pair. Figure 5 shows the calculating process and an illustrative example. The reason we employ a collocation like “noun-verb” or “noun-adjective” as our starting words to generate utterances lies in that we believe this is the most simple but reliable way to grab a meaningful snippet containing an inherent focus from the Web. We will give more detailed description on utterance generation from starting words in Section 4. Finally, three topic-word-pair candidates having the largest degree of association are extracted from the Utterance Window in this manner, and handed over to Multiple-word Generating Module for utterance generation.

3.4 Discussions

We must note that the procedure described above is taken only after the conversation has continued for a certain period of time, i.e., in case we already have enough user-utterances to form a complete Utterance Window. At the initial stage of the conversation, or in other words, while we don't have n or more user-utterances yet, the

original algorithm keeps working and only one topic word will be extracted in the Planning Block, and then passed on to the Generating Block to generate a new utterance sentence starting from this single topic word, as done in the previous work [8] and shown in Figure 1. This mechanism conforms to the nature of human conversations and works fine in two points: the change in reply time and the shift in conversation's focus.

When a user has just begun chatting with the computer, it takes longer for the system to find a topic word as the original Planning Block depends on Web searching via Google API to accomplish this task. Web searching is generally time-consuming and yields more random searching results at the same time. This happens to be the case of human conversations. When two strangers meet for the first time, they tend to take time to carefully find some topics that are not impolite, and sometimes even meaningless to talk about with the counterpart.

On the other hand, when the conversation between the user and the computer has continued for a period of time to accumulate enough user-utterances to compose an Utterance Window, the system automatically switches the topic-word-extraction task to the new Extracting Block. The process carried out here in most cases shouldn't be much time-consuming, but more accurate and more context-dependent to capture the real focus of the current conversation. Also, a similar situation could be observed in interpersonal conversations where two strangers begin to become familiar with each other and hence tend to speed up their conversation with more constant topics.

In this manner, we try to make the system behave like a real human being, and produce a more natural conversation transition between the user and the computer.

4 Utterance Generation

Using the topic-word pair extracted in Section 3.3, the system tries to search the Twitter for snippets that contain the two topic words through Twitter API, and then generates an utterance employing a Two-starting-word style Markov connection.

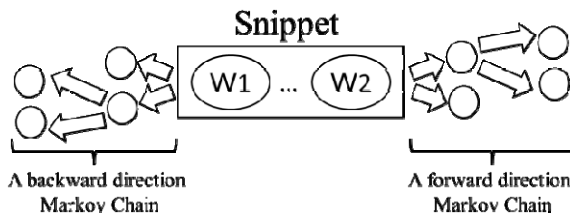


Fig. 6. Processing Schema of Multiple-word Generating Module

We use $W1$ to denote the left-side word, i.e., the noun, and $W2$ for the right-side word (the verb or adjective) in the topic-word pair. $W1$ and $W2$ act as a searching query in Twitter API, and bring back a number of searching results containing snippets in the form of $\dots W1\dots W2\dots$. Then the system selects a snippet from all the

searching results and cut off both the left-hand texts attached to W1 and the right-hand to W2 to leave the part of W1...W2 only. At last, an utterance sentence is generated by taking W1...W2 as the starting point, and extending to both directions based on the bi-directional Markov dictionaries simultaneously, until encountering the beginning signal in the left and the ending signal in the right. The Process schema is shown in Figure 6.

The reason we prefer two-starting-word style to one-starting-word version is that we think a snippet containing two mutually associated words appears more meaningful and more conspicuous to make the focus of the sentence clear.

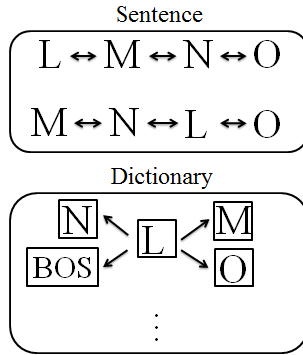


Fig. 7. Schema of Bi-directional Markov Dictionary

During the Twitter searching procedure, we keep adding 2-grams to a hash-dictionary called Bi-directional Markov Dictionary. The dictionary is composed of two parts: a forward-direction dictionary, and a backward-direction dictionary. The upper part in Figure 7 shows some 2-gram examples extracted from sentences, and the lower part illustrates the dictionary entries coming out from the n-grams. That way we try to speed up the utterance generation task by using more and more data from the local database, rather than fetching data from the Web all the time. In case we could not find any snippets using the topic-word pair with the highest mutual association score, the second and the third topic-word pair will be taken to have another trials in turn.

5 Evaluations

We have conducted a set of experiments to examine the effectiveness of the Utterance Window and the performance of the Multiple-word generating module. In this section, we describe the experimental method and discuss their results.

5.1 Subjective Evaluation

We have built three kinds of prototypes based on the algorithms described in Section 2, 3, and 4 where n (the total number of user-utterances in the Utterance Window) is

set to be 3, 5, and 10 respectively. Also we have prepared another prototype developed in the preceding work. Then a user keeps chatting with each of the four systems, and randomly selects a snippet with the same length from the conversation history with each system.

A subjective assessment is carried out then with three subjects who haven't involved in this work so far. Given some simple instructions, the subjects are told to evaluate each of the four systems in two points: Association between Utterances, and Utterance Focus. The former evaluation item indicates the association between continuous utterances, i.e., how good has the conversation topic transitioned? A rank-order assessment is supposed to be made with 1 indicating the best system, and 4 indicating the worst system which means the conversation flow is quite bad in topic transition. The latter evaluation item, Utterance Focus, evaluates how good the quality of an utterance sentence is without taking consideration of other utterances. Similarly, we give a rank order from 1 to 4 to each system to evaluate the quality of the utterance sentence.

Average rank-orders of all subjects are shown in Table 1. From the results we can see that our system with a 5-user-utterance window behaves best in general. This proves the effectiveness of our strategy to cope with the weak association between continuous user-utterances. However, the result is not as good as we have expected. Our new system could not outperform the previous work in Utterance Focus. This seems to come from the nature of our two-starting-word method and the unlimited length of the generated computer-utterance. A sentence tends to be out of control and the topic blurs when it is getting too long. A possible way to improve this situation is probably limit the generated utterance within a certain length.

Table 1. Questionnaire Results

	Preceding System	Our System		
		<i>n</i> =3	<i>n</i> =5	<i>n</i> =10
Association between Utterances	3.0	3.3	1.0	2.7
Utterance Focus	1.3	3.7	2.3	2.7
Average	2.2	3.5	1.7	2.7

5.2 Evaluation on Topic-Word Pair Selection

Another experiment has been conducted to validate the effectiveness of association calculation between topic words defined in Section 3.2 and 3.3. We randomly select 30 Utterance Windows from the 5-utterance-window prototype, get all the three topic-word candidates from each window, and record their mutual association calculated in Section 3.3. Then we compute another kind of association based on Web search using Google API with the formula below.

$$\frac{f(W_a, W_b)}{f(W_a)f(W_b)}$$

Here, $f(W_a)$ indicates the hit number of web pages containing the topic word W_a , and $f(W_a, W_b)$ indicates that of pages containing W_a and W_b simultaneously. We believe that the topic-word pair candidate with a high mutual association based on Web should be appropriate to act as the two starting points in a new sentence, and hence expect to examine the effectiveness of our definitions in Section 3 by checking how similar the Utterance-Window based rank order is to that based on the Web.

Table 2. Rank Orders in Different Association-Estimation

Our Sys- tem	Web-based method
1	1.9
2	2.0
3	2.2

Table 2 shows the results. For instance, the number 1.9 in the upper-right cell indicates the average rank order of Web-based mutual associations between all the topic words with the highest Utterance-Window based mutual associations. Results here show that our method conforms exactly to the Web-based method which indicates the effectiveness of our system.

6 Conclusion

In this paper, we propose a major improving strategy to a previously developed non-task-oriented conversation system. Specifically, we introduce a concept called Utterance Window through which we expect to strengthen the association between continuous utterances, and employ a Two-starting-word Markov connection instead of the original Single-starting-word manner, to cope with the ease of losing focus of the current utterance. Also, we try to keep track of user's interests and reflecting them in the process of topic-word extraction and utterance generation as well.

We have shown the effectiveness of our method through a set of experiments. However, the result is not as good as we have expected, especially in keeping Utterance Focus from spreading around too widely. This seems to be due to the nature of our two-starting-word method which tends to generate comparatively long computer-utterances. We are planning to incorporate some restrictive rules to improve this situation by limiting the total number of words or characters contained in the generated utterance.

References

1. Fujimoto, E., Takanashi, K., Kono, Y., Kidoe, M.: An Analysis of Topic Changes in Free Conversation Using Conceptual Relations. In: Proceedings of the 18th Annual Conference of the Japanese Society for Artificial Intelligence, 2G3-01 (2004) (in Japanese)
2. Saito, T., Hirota, K., Hoshino, J.: Utterance and Small Talk Model between Characters by Using Web Information. SIG notes, NL-2007(181), Information Processing Society of Japan, pp. 53–58 (2007) (in Japanese)
3. Higuchi, S., Rzepka, R., Araki, K.: A Casual Conversation System Using Modality and Word Associations Retrieved from the Web. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 382–390 (2008)
4. Yoshioka, K., Yoshimura, E., Watabe, H., Kawaoka, T.: Computer Conversational Processing System Based on Individuality and Personal Preference Information. In: Proceedings of FIT 2008, vol. (2), pp. 289–290 (2008)
5. Song, X., Maeda, K., Kunimasa, H., Toyota, H., Han, D.: Topic Control in A Free Conversation System. In: Proceedings of the 2009 IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 529–534 (2009)
6. Han, D., Song, X., Maeda, K.: Topic Presentation for a Free Conversation System Based on the Web Texts. *International Journal of Digital Content Technology and its Applications* 4(3), 7–14 (2010)
7. Takahashi, M., Rzepka, R., Araki, K.: Improving Utterance Generation Method Based on Word Ngrams. In: Joint Convention Record, the Hokkaido Chapters of the Institute of Electrical and Information Engineers, vol. 112 (2009) (in Japanese)
8. Han, D., Kinoshita, Y., Fukuchi, R., Kousaki, T.: Utterance Generation Using Twitter Replying Sentences and Character Assignment. *International Journal of Digital Content Technology and its Applications* 5(10), 119–126 (2011)
9. Kurohahsi, S., Shiraki, N., Nagao, M.: A Method for Detecting Important Descriptions of a Word Based on Its Density Distribution in Text. *Transactions of Information Processing Society of Japan* 38(4), 845–854 (1997) (in Japanese)
10. Satake, S., Kawashima, H., Imai, M.: Contents Selection Method Based on A User Interest on the Interactive News Announcer Robot. Technical Report of IEICE, DE-2005(50), pp. 119–124 (2005) (in Japanese)
11. Church, K., Hanks, P.: Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1), 22–29 (1990)

Author Index

- Affi, Haithem 40
Aikawa, Takako 1
Aliwy, Ahmed H. 168
- Barrault, Loïc 40
Beaufort, Richard 300
Brixtel, Romain 64
Brognaux, Sandrine 300
- Cardey-Greenfield, Sylviane 118, 138
Chen, Shaoyu 11
- Doucet, Antoine 64
Drugman, Thomas 300
- Endo, Tsutomu 278
- Feuto Njonko, Paul Brillant 118
Fukazawa, Yusuke 262
Fukuhara, Tomohiro 239
- García Flores, Jorge J. 180
Ghayoomi, Masood 126
Gołuchowski, Konrad 192
Greenfield, Peter 118
- Hamada, Shogo 268
Han, Chia Y. 228
Han, Dongli 322
Hatamoto, Norinobu 268
- Imabuchi, Shohei 312
Isahara, Hitoshi 1, 23
- Jarzębowski, Przemysław 198
Jin, Gan 52
- Kampeera, Wannachai 138
Kando, Noriko 239
Karapetsas, Eleftherios 262
Kathuria, Pulkit 210
Khatseyeva, Natallia 52
Kiyota, Yoji 239
Kurosawa, Yoshiaki 268
- Le, Tho Thi Ngoc 222
Lee, Samuel Sangkon 228
- Lejeune, Gaël 64
Lucas, Nadine 64
Lytinen, Steve 289
- Maeda, Hiroshi 278
Matsumoto, Tadahiro 11
Mera, Kazuya 268
Murakami, Jin'ichi 28
- Nakagawa, Hiroshi 239
Nguyen, Le Minh 76, 108, 222, 250
Nishio, Yusuke 322
- Ogata, Takashi 312
Ota, Jun 262
- Przepiórkowski, Adam 144, 192, 198
- Qu, Jian 76
- Raison, Kevin 289
Roekhaut, Sophie 300
- Sadat, Fatiha 88
Sakata, Jun 28
Sanyal, Sudip 97
Schwenk, Holger 40
Shimada, Kazutaka 278
Shimazu, Akira 76, 222, 250
Shirai, Kiyooki 210
Srivastava, Jyoti 97
- Takahashi, Yusuke 239
Takezawa, Toshiyuki 268
Tokuhisa, Masato 28
Tomuro, Noriko 289
Tu, Dao Ngoc 108
Turner, William 180
- Utsuro, Takehito 239
- Van Nguyen, Vinh 108
Vuong, Hoai-Thu 108
- Xuan Bach, Ngo 250
- Yamamoto, Kentaro 1
Yoshioka, Masaharu 239

Zaborowski, Bartosz 144
Zagal, Jose P. 289
Zhao, Yue 180

Zhu, Dandan 262
Zimina, Elizaveta 156
Zweigenbaum, Pierre 180