ACL HLT 2011

49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies

Proceedings of Student Session

19-24 June 2011
Portland, Oregon, USA

Production and Manufaturing by

Omnipress, Inc.
2600 Anderson Street
Madison, WI 53704
USA

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-932432-89-3

Introduction

Welcome to the ACL-HLT 2011 Student Session. This year, we have two separate submission types: Research Papers and Thesis Proposals. We accepted 22 out of the 57 submissions received from a wide variety of countries. Of the accepted submissions, 18 were Research Papers and 4 were Thesis Proposals. Each accepted paper is to be presented as a poster during the ACL-HLT 2011 Poster Session and assigned a mentor to provide specific feedback and guidance to our authors.

The overall quality of the submissions were high and we thank our Program Committee for their excellent feedback and reviews. We also thank our Faculty Advisors, Miles Osborne and Thamar Solorio, for their patience and guidance. We would also like to thank the entire ACL-HLT 2011 Organizing Committee, especially Dekang Lin, Brian Roark, Nate Bodenstab, Guodong Zhou, and Priscilla Rasmussen. All presenters were offered conference registration and travel support, thanks to generous support from the U.S. National Science Foundation (Award Number 1102435), the ACL Walker Student Fund, Google, and the European Association of Computational Linguistics. Finally, our highest gratitude goes to all the students who submitted to the Student Session. Thank you.

Matt Huenerfauth, *The City University of New York, USA*
Michael Johnston, *AT&T Labs, USA*
Udo Kruschwitz, *University of Essex, UK*
Giridhar Kumaran, *Microsoft, USA*
Victor Lavrenko, *University of Edinburgh, UK*
Chin-Yew Lin, *Microsoft, USA*
Adam Lopez, *Johns Hopkins University, USA*
David McClosky, *Stanford University, USA*
Donald Metzler, *Univerity of Southern California, USA*
Taniya Mishra, *AT&T Labs, USA*
Saif Mohammad, *National Research Council Canada, Canada*
Sebastian Pado, *University of Stuttgart, Germany*
Katerina Pastra, *Institute for Language and Speech Processing, Greece*
Pavel Pecina, *Dublin City University, Ireland*
Hema Raghavan, *Yahoo, USA*
Jan Romportl, *University of West Bohemia, Czech Republic*
Helmut Schmid, *University of Stuttgart, Germany*
Benjamin Snyder, *University of Wisconsin-Madison, USA*
Veselin Stoyanov, *Johns Hopkins University, USA*
Yi Su, *Nuance, USA*
Joel Tetreault, *Educational Testing Service, USA*
Lucy Vanderwende, *Microsoft, USA*
Yorick Wilks, *University of Sheffield, UK*
Fan Yang, *Nuance, USA*
Alexander Yates, *Temple University, USA*
Emine Yilmaz, *Microsoft, USA*
Daniel Zeman, *Charles University in Prague, Czech Republic*

Student Co-Chairs:

Sasa Petrovic, University of Edinburgh
Emily Pitler, University of Pennsylvania
Ethan Selfridge, Oregon Health & Science University

Faculty Advisors:

Miles Osborne, University of Edinburgh
Thamar Solorio, University of Alabama at Birmingham

Table of Contents

Conference Program

Monday, June 20, 2011

(12:00-2:00)Student Lunch

(6:00-8:30)Poster Session(Student Session)

Word Alignment Combination over Multiple Word Segmentation
Ning Xi, Guangchao Tang, Boyuan Li and Yinggong Zhao

Sentence Ordering Driven by Local and Global Coherence for Summary Generation
Renxian Zhang

Pre- and Postprocessing for Statistical Machine Translation into Germanic Languages
Sara Stymne

Exploring Entity Relations for Named Entity Disambiguation
Danuta Ploch

Extracting and Classifying Urdu Multiword Expressions
Annette Hautli and Sebastian Sulger

A Latent Topic Extracting Method based on Events in a Document and its Application
Risa Kitajima and Ichiro Kobayashi

Syntax-based Statistical Machine Translation using Tree Automata and Tree Transducers
Daniel Emilio Beck

ConsentCanvas: Automatic Texturing for Improved Readability in End-User License Agreements
Oliver Schneider and Alex Garnett

Disambiguating temporal-contrastive connectives for machine translation
Thomas Meyer

PsychoSentiWordNet
Amitava Das

Optimistic Backtracking - A Backtracking Overlay for Deterministic Incremental Parsing
Gisle Ytrestøl

An Error Analysis of Relation Extraction in Social Media Documents
Gregory Brown

Monday, June 20, 2011(continued)

Effects of Noun Phrase Bracketing in Dependency Parsing and Machine Translation
Nathan Green

Towards a Framework for Abstractive Summarization of Multimodal Documents
Charles Greenbacker

Sentiment Analysis of Citations using Sentence Structure-Based Features
Awais Athar

Combining Indicators of Allophony
Luc Boruta

Turn-Taking Cues in a Human Tutoring Corpus
Heather Friedberg

Predicting Clicks in a Vocabulary Learning System
Aaron Michelony

Exploiting Morphology in Turkish Named Entity Recognition System
Reyyan Yeniterzi

Social Network Extraction from Texts: A Thesis Proposal
Apoorv Agarwal

Automatic Headline Generation using Character Cross-Correlation
Fahad Alotaiby

K-means Clustering with Feature Hashing
Hajime Senuma

Word Alignment Combination over Multiple Word Segmentation

Ning Xi, Guangchao Tang, Boyuan Li, Yinggong Zhao
State Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology,
Nanjing University, Nanjing, 210093, China
{xin,tanggc,liby,zhaoyg}@nlp.nju.edu.cn

Abstract

In this paper, we present a new word alignment combination approach on language pairs where one language has no explicit word boundaries. Instead of combining word alignments of different models (Xiang et al., 2010), we try to combine word alignments over multiple monolingually motivated word segmentation. Our approach is based on link confidence score defined over multiple segmentations, thus the combined alignment is more robust to inappropriate word segmentation. Our combination algorithm is simple, efficient, and easy to implement. In the Chinese-English experiment, our approach effectively improved word alignment quality as well as translation performance on all segmentations simultaneously, which showed that word alignment can benefit from complementary knowledge due to the diversity of multiple and monolingually motivated segmentations.

1 Introduction

Word segmentation is the first step prior to word alignment for building statistical machine translations (SMT) on language pairs without explicit word boundaries such as Chinese-English. Many works have focused on the improvement of word alignment models. (Brown et al., 1993; Haghighi et al., 2009; Liu et al., 2010). Most of the word alignment models take single word segmentation as input. However, for languages such as Chinese, it is necessary to segment sentences into appropriate words for word alignment.

A large amount of works have stressed the impact of word segmentation on word alignment. Xu et al. (2004), Ma et al. (2007), Chang et al. (2008), and Chung et al. (2009) try to learn word segmentation from bilingually motivated point of view; they use an initial alignment to learn word segmentation appropriate for SMT. However, their performance is limited by the quality of the initial alignments, and the processes are time-consuming. Some other methods try to combine multiple word segmentation at SMT decoding step (Xu et al., 2005; Dyer et al., 2008; Zhang et al., 2008; Dyer et al., 2009; Xiao et al., 2010). Different segmentations are yet independently used for word alignment.

Instead of time-consuming segmentation optimization based on alignment or postponing segmentation combination late till SMT decoding phase, we try to combine word alignments over multiple monolingually motivated word segmentation on Chinese-English pair, in order to improve word alignment quality and translation performance for all segmentations. We introduce a tabular structure called word segmentation network (WSN for short) to encode multiple segmentations of a Chinese sentence, and define skeleton links (SL for short) between spans of WSN and words of English sentence. The confidence score of a SL is defined over multiple segmentations. Our combination algorithm picks up potential SLs based on their confidence scores similar to Xiang et al. (2010), and then projects each selected SL to link in all segmentation respectively. Our algorithm is simple, efficient, easy to implement, and can effectively improve word alignment quality on all segmentations simultaneously, and alignment errors caused

Proceedings of the ACL-HLT 2011 Student Session, pages 1–5,
Portland, OR, USA 19-24 June 2011.

by inappropriate segmentations from single segmenter can be substantially reduced.

Two questions will be answered in the paper: 1) how to define the link confidence over multiple segmentations in combination algorithm? 2) According to Xiang et al. (2010), the success of their word alignment combination of different models lies in the complementary information that the candidate alignments contain. In our work, are multiple monolingually motivated segmentations complementary enough to improve the alignments?

The rest of this paper is structured as follows: WSN will be introduced in section 2. Combination algorithm will be presented in section 3. Experiments of word alignment and SMT will be reported in section 4.

2 Word Segmentation Network

We propose a new structure called word segmentation network (WSN) to encode multiple segmentations. Due to space limitation, all definitions are presented by illustration of a running example of a sentence pair:

下雨路滑 (xia-yu-lu-hua)
Road is slippery when raining

We first introduce *skeleton segmentation*. Given two segmentation S_1 and S_2 in Table 1, the word boundaries of their skeleton segmentation is the union of word boundaries (marked by "/") in S_1 and S_2.

	Segmentation
S_1	下 / 雨 / 路滑
S_2	下雨 / 路 / 滑
skeleton	下 / 雨 / 路 / 滑

Table 1: The skeleton segmentation of two segmentations S1 and S2.

The WSN of S_1 and S_2 is shown in Table 2. As is depicted, line 1 and 2 represent words in S_1 and S_2 respectively, line 3 represents skeleton words. Each column, or span, comprises a skeleton word and words of S_1 and S_2 with the skeleton word as their morphemes at that position. The number of columns of a WSN is equal to the number of skeleton words. It should be noted that there may be words covering two or more spans, such as "路滑" in S_1, because the word "路滑" in S_1 is split into two words "路" and "滑" in S_2.

S_1	下 $_1$	雨 $_2$	路滑 $_3$	
S_2	下雨 $_1$		路 $_2$	滑 $_3$
skeleton	下 $_1$	雨 $_2$	路 $_3$	滑 $_4$

Table 2: The WSN of Table 1. Subscripts indicate indexes of words.

The skeleton word can be projected onto words in the same span in S_1 and S_2. For clarity, words in each segmentation are indexed (1-based), for example, "路滑" in S_1 is indexed by 3. We use a projection function $\delta_k(j)$ to denote the index of the word onto which the *j*-th skeleton word is projected in the *k*-th segmentation, for example, $\delta_1(4) = 3$ and $\delta_2(3) = 2$.

In the next, we define the links between spans of the WSN and English words as skeleton links (SL), the subset of all SLs comprise the skeleton alignment (SA). Figure 1 shows an SA of the example.

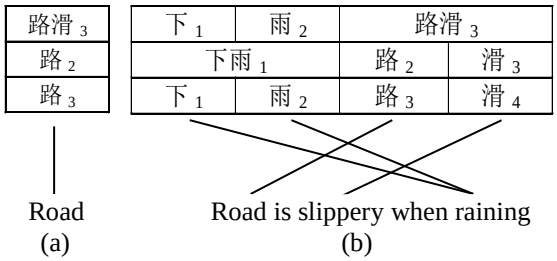

Figure 1: An example alignment between WSN in Table 2 and English sentence "Road is slippery when raining". (a) skeleton link; (b) skeleton alignment.

Each span of the WSN comprises words from different segmentations (Figure 1a), which indicates that the confidence score of a SL can be defined over words in the same span. By projection function, a SL can be projected onto the link for each segmentation. Therefore, the problem of combining word alignment over different segmentations can be transformed into the problem of selecting SLs for SA first, and then project the selected SLs onto links for each segmentation respectively.

3 Combination Algorithm

Given k alignments a_k over segmentations s_k respectively $(k = 1, \dots, K)$, and (C, E) is the pair

of the Chinese WSN and its parallel English sentence. Suppose A_{ij} is the SL between the *j*-th span c_j and *i*-th English word e_i, a_{ij}^k is the link between the *j*-th Chinese word c_j^k in s_k and e_i. Inspired by Huang (2009), we define the confidence score of each SL as follows

$$C(A_{ij}|C,E) = \sum_{k=1}^{K} w_i * c(a_{i\delta_k(j)}^k|C,E) \quad (1)$$

where $c(a_{i\delta_k(j)}^k|C,E)$ is the confidence score of the link $a_{i\delta_k(j)}^k$, defined as

$$c(a_{i\delta_k(j)}^k|C,E) = \sqrt{q_{c2e}(a_{i\delta_k(j)}^k|C,E) * q_{e2c}(a_{i\delta_k(j)}^k|C,E)} \quad (2)$$

where c-to-e link posterior probability is defined as

$$q_{c2e}(a_{i\delta_k(j)}^k|C,E) = \frac{p_k(e_i|c_{\delta_k(j)}^k)}{\sum_{i'=1}^{I} p_k(e_{i'}|c_{\delta_k(j)}^k)} \quad (3)$$

and *I* is the length of E. E-to-c link posterior probability $q_{e2c}(a_{i\delta_k(j)}^k|C,E)$ can be defined similarly,

Our alignment combination algorithm is as follows.

1. Build WSN for Chinese sentence.
2. Compute the confidence score for each SL based on Eq. (1). A SL A_{ij} gets a vote from a_k if $a_{i\delta_k(j)}^k$ appears in a_k $(k = 1, \dots, K)$. Denote the set of all SLs getting at least one vote by B_0.
3. All SLs in B_0 are sorted in descending order and evaluated sequentially. A SL A_{ij} is included if its confidence score is higher than a tunable threshold α, and one of the following is true[1]:
   - Neither c_j nor e_j is aligned so far;
   - c_j is not aligned and its left or right neighboring word is aligned to e_j so far;
   - e_j is not aligned and its left or right neighboring word is aligned to c_j so far.
4. Repeat 3 until no more SLs can be included. All included SLs comprise B_1.
5. Map SLs in B_1 on each s_k to get *k* new alignments a'_k respectively, i.e. $a'_k = \{a_{i\delta_k(j)}^k | A_{ij} \in B_1\}$[2]$(k = 1, \dots, K)$. For each k, we sort all links in a'_k in ascending order and evaluated them sequentially. Compare a'_k and a_k, A link a'^k_{ij} is removed from a'_k if it is not appeared in a_k, and one of the following is true:
   - both c_j^k and e_j are aligned in a'_k;
   - There is a word which is neither left nor right neighboring word of e_j but aligned to c_j^k in a'_k;
   - There is a word which is neither left nor right neighboring word of c_j^k but aligned to e_j in a'_k.

The heuristic in step 3 is similar to Xiang et al. (2010), which avoids adding error-prone links. We apply the similar heuristic again in step 5 in each a'_k $(k = 1, \dots, K)$ to delete error-prone links. The weights in Eq. (1) and α can be tuned in a hand-aligned dataset to maximize word alignment F-score on any a'_k with hill climbing algorithm. Probabilities in Eq. (2) and Eq. (3) can be estimated using GIZA.

4 Experiment

4.1 Data

Our training set contains about 190K Chinese-English sentence pairs from LDC2003E14 corpus. The NIST'06 test set is used as our development set and the NIST'08 test set is used as our test set. The Chinese portions of all the data are preprocessed by three monolingually motived segmenters respectively. These segmenters differ in either training method or specification, including ICTCLAS (I)[3], Stanford segmenters with CTB (C) and PKU (P) specifications[4] respectively. We used a phrase-based MT system similar to (Koehn et al., 2003), and generated two baseline alignments using GIZA++ enhanced by *gdf* heuristics (Koehn et al., 2003) and a linear discriminative word alignment model (DIWA) (Liu et al., 2010) on training set with the three segmentations respectively. A 5-gram language model trained from the Xinhua portion of Gigaword corpus was used. The decoding weights were optimized with Minimum Error Rate Training (MERT) (Och, 2003). We used the hand-aligned set of 491 sentence pairs in Haghighi et al. (2009), the first 250 sentence pairs were used to tune the weights in Eq. (1), and the other 241 were

[1] SLs getting *K* votes are forced to be included without further examination.

[2] Two or more SLs in B_1 may be projected onto one links in a'_k, in this case, we keep only one in a'_k.

[3] http://www.ictclas.org/

[4] http://nlp.stanford.edu/software/segmenter.shtml

Figure 2: Two examples (left and right respectively) of word alignment on segmentation C. Baselines (DIWA) are in the top half, combined alignments are in the bottom half. The solid line represents the correct link while the dashed line represents the bad link. Each word is enclosed in square brackets.

used to measure the word alignment quality. Note that we adapted the Chinese portion of this hand-aligned set to segmentation C.

4.2 Improvement of Word Alignment

We first evaluate our combination approach on the hand-aligned set (on segmentation C). Table 3 shows the precision, recall and F-score of baseline alignments and combined alignments.

As shown in Table 3, the combination alignments outperformed the baselines (setting C) in all settings in both GIZA and DIWA. We notice that the higher F-score is mainly due to the higher precision in GIZA but higher recall in DIWA. In GIZA, the result of C+I and C+P achieve 8.4% and 9.5% higher F-score respectively, and both of them outperformed C+P+I, we speculate it is because GIZA favors recall rather than DIWA, i.e. GIZA may contain more bad links than DIWA, which would lead to more unstable F-score if more alignments produced by GIZA are combined, just as the poor precision (69.68%) indicated. However, DIWA favors precision than recall (this observation is consistent with Liu et al. (2010)), which may explain that the more diversified segmentations lead to better results in DIWA.

	GIZA			DIWA		
setting	**P**	**R**	**F**	**P**	**R**	**F**
C	61.84	84.99	71.59	83.12	78.88	80.94
C+P	80.16	79.80	79.98	84.15	79.41	81.57
C+I	82.96	79.28	81.08	84.41	81.69	83.03
C+I+P	69.68	85.17	77.81	83.38	82.98	83.18

Table 3: Alignment precision, recall and F-score. C: baseline, C+I: Combination of C and I.

Figure 2 gives baseline alignments and combined alignments on two sentence pairs in the training data. As can be seen, alignment errors caused by inappropriate segmentations by single segmenter were substantially reduced. For example, in the second example, the word "香港特别行政区 hksar" appears in segmentation I of the Chinese sentence, which benefits the generation of the three correct links connecting for words "香港" ,"特别", "行政区" respectively in the combined alignment.

4.3 Improvement in MT performance

We then evaluate our combination approach on the SMT training data on all segmentations. For efficiency, we just used the first 50k sentence pairs of the aligned training corpus with the three segmentations to build three SMT systems respectively. Table 4 shows the BLEU scores of baselines and combined alignment (C+P+I, and then projected onto C, P, I respectively). Our approach achieves improvement over baseline alignments on all segmentations consistently, without using any lattice decoding techniques as Dyer et al. (2009). The gain of translation performance purely comes from improvements of word alignment on all segmentations by our proposed word alignment combination.

	GIZA		DIWA	
Segmentation	**B**	**Comb**	**B**	**Comb**
C	19.77	20.9	20.18	20.71
P	20.5	21.16	20.41	21.14
I	20.11	21.14	20.46	21.30

Table 4: Improvement in BLEU scores. B:Baseline alignment, Comb: Combined alignment.

5 Conclusion

We evaluated our word alignment combination over three monolingually motivated segmentations on Chinese-English pair. We showed that the combined alignment significantly outperforms the baseline alignment with both higher F-score and higher BLEU score on all segmentations. Our work also proved the effectiveness of link confidence score in combining different word alignment models (Xiang et al., 2010), and extend it to combine word alignments over different segmentations.

Xu et al. (2005) and Dyer et al. (2009) combine different segmentations for SMT. They aim to achieve better translation but not higher alignment quality of all segmentations. They combine multiple segmentations at SMT decoding step, while we combine segmentation alternatives at word alignment step. We believe that we can further improve the performance by combining these two kinds of works. We also believe that combining word alignments over both monolingually motivated and bilingually motivated segmentations (Ma et al., 2009) can achieve higher performance.

In the future, we will investigate combining word alignments on language pairs where both languages have no explicit word boundaries such as Chinese-Japanese.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61003112, and the National Fundamental Research Program of China (2010CB327903). We would like to thank Xiuyi Jia and Shujie Liu for useful discussions and the anonymous reviewers for their constructive comments.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Peitra, Robert L. Mercer. 1993. *The Mathematics of statistical machine translation: parameter estimation*. *Computational Linguistics*, 19(2):263-311.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. *Optimizing Chinese word segmentation for machine translation performance*. In *Proceedings of* third workshop on SMT, Pages:224-232.

Tagyoung Chung and Daniel Gildea. 2009. *Unsupervised tokenization for machine translation*. In *Proceedings of* EMNLP, Pages:718-726.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. *Generalizing word lattice translation*. In *Proceedings of* ACL, Pages:1012-1020.

Christopher Dyer. 2009. *Using a maximum entropy model to build segmentation lattices for mt*. In *Proceedings of* NAACL, Pages:406-414.

Franz Josef Och. 2003. *Minimum error rate training in statistical machine translation*. In *Proceedings of* ACL, Pages:440-447.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. *Better word alignments with supervised ITG models*. In *Proceedings of* ACL, Pages: 923-931.

Fei Huang. 2009. Confidence measure for word alignment. In *Proceedings of* ACL, Pages:932-940.

Philipp Koehn, Franz Josef Och and Daniel Marcu. 2003. *Statistical phrase-based translation.* In *Proceedings of* HLT-NAACL, Pages:48-54.

Yang Liu, Qun Liu, Shouxun Lin. 2010. *Discriminative word alignment by linear modeling*. Computational Linguistics, 36(3):303-339.

Yanjun Ma, Nicolas Stroppa, and Andy Way. 2007. *Bootstrapping word alignment via word packing*. In *Proceedings of* ACL, Pages:304-311.

Yanjun Ma and Andy Way. 2009. *Bilingually motivated domain-adapted word segmentation for statistical machine translation.* In *Proceedings of* EACL, Pages:549-557.

Bing Xiang, Yonggang Deng, and Bowen Zhou. 2010. *Diversify and combine: improving word alignment for machine translation on low-resource languages.* In *Proceedings of* ACL, Pages:932-940.

Xinyan Xiao, Yang Liu, Young-Sook Hwang, Qun Liu, Shouxun Lin. 2010. *Joint tokenization and translation*. In *Proceedings of* COLING, Pages:1200-1208.

Jia Xu, Richard Zens, and Hermann Ney. 2004. *Do we need Chinese word segmentation for statistical machine translation?* In *Proceedings of* the ACL SIGHAN Workshop, Pages: 122-128.

Jia Xu, Evgeny Matusov, Richard Zens, and Hermann Ney. 2005. *Integrated Chinese word segmentation in statistical machine translation*. In *Proceedings of* IWSLT.

Ruiqiang Zhang, Keiji Yasuda, and Eiichiro Sumita. 2008. *Improved statistical machine translation by multiple Chinese word segmentation.* In *Proceedings of* the Third Workshop on SMT, Pages:216-223.

Sentence Ordering Driven by Local and Global Coherence for Summary Generation

Renxian Zhang
Department of Computing
The Hong Kong Polytechnic University
csrzhang@comp.polyu.edu.hk

Abstract

In summarization, sentence ordering is conducted to enhance summary readability by accommodating text coherence. We propose a grouping-based ordering framework that integrates local and global coherence concerns. Summary sentences are grouped before ordering is applied on two levels: group-level and sentence-level. Different algorithms for grouping and ordering are discussed. The preliminary results on single-document news datasets demonstrate the advantage of our method over a widely accepted method.

1 Introduction and Background

The canonical pipeline of text summarization consists of topic identification, interpretation, and summary generation (Hovy, 2005). In the simple case of extraction, topic identification and interpretation are conflated to sentence selection and concerned with summary informativeness. In comparison, summary generation addresses summary readability and a frequently discussed generation technique is sentence ordering.

It is implicitly or explicitly stated that sentence ordering for summarization is primarily driven by coherence. For example, Barzilay et al. (2002) use lexical cohesion information to model local coherence. A statistical model by Lapata (2003) considers both lexical and syntactic features in calculating local coherence. More globally biased is Barzilay and Lee's (2004) HMM-based content model, which models global coherence with word distribution patterns.

Whilst the above models treat coherence as lexical or topical relations, Barzilay and Lapata (2005, 2008) explicitly model local coherence with an entity grid model trained for optimal syntactic role transitions of entities.

Although coherence in those works is modeled in the guise of "lexical cohesion", "topic closeness", "content relatedness", etc., few published works simultaneously accommodate coherence on the two levels: local coherence and global coherence, both of which are intriguing topics in text linguistics and psychology. For sentences, local coherence means the well-connectedness between adjacent sentences through lexical cohesion (Halliday and Hasan, 1976) or entity repetition (Grosz et al., 1995) and global coherence is the discourse-level relation connecting remote sentences (Mann and Thompson, 1995; Kehler, 2002). An abundance of psychological evidences show that coherence on both levels is manifested in text comprehension (Tapiero, 2007). Accordingly, an apt sentence ordering scheme should be driven by such concerns.

We also note that as sentence ordering is usually discussed only in the context of multi-document summarization, factors other than coherence are also considered, such as time and source sentence position in Bollegala et al.'s (2006) "agglomerative ordering" approach. But it remains an open question whether sentence ordering is non-trivial for single-document summarization, as it has long been recognized as an actual strategy taken by human summarizers (Jing, 1998; Jing and McKeown, 2000) and acknowledged early in work on sentence ordering for multi-document summarization (Barzilay et al., 2002).

In this paper, we outline a grouping-based sentence ordering framework that is driven by the concern of local and global coherence. Summary sentences are grouped according to their conceptual relatedness before being ordered on two levels: group-level ordering and sentence-level ordering, which capture global coherence and local coherence in an integrated model. As a preliminary study, we applied the framework to single-

Proceedings of the ACL-HLT 2011 Student Session, pages 6–11,
Portland, OR, USA 19-24 June 2011.

document summary generation and obtained interesting results.

The main contributions of this work are: (1) we stress the need to channel sentence ordering research to linguistic and psychological findings about text coherence; (2) we propose a grouping-based ordering framework that integrates both local and global coherence; (3) we find in experiments that coherence-driven sentence ordering improves the readability of single-document summaries, for which sentence ordering is often considered trivial.

In Section 2, we review related ideas and techniques in previous work. Section 3 provides the details of grouping-based sentence ordering. The preliminary experimental results are presented in Section 4. Finally, Section 5 concludes the whole paper and describes future work.

2 Grouping-Based Ordering

Our ordering framework is designed to capture both local and global coherence. Globally, we identify related groups among sentences and find their relative order. Locally, we strive to keep sentence similar or related in content close to each other within one group.

2.1 Sentence Representation

As summary sentences are isolated from their original context, we retain the important content information by representing sentences as concept vectors. In the simplest case, the "concept" is equivalent to content word. A drawback of this practice is that it considers every content word equally contributive to the sentence content, which is not always true. For example, in the news domain, entities realized as NPs are more important than other concepts.

To represent sentences as entity vectors, we identify both common entities (as the head nouns of NPs) and named entities. Two common entities are equivalent if their noun stems are identical or synonymous. Named entities are usually equated by identity. But in order to improve accuracy, we also consider: 1) structural subsumption (one is part of another); 2) hypernymy and holonymy (the named entities are in a superordinate-subordinate or part-whole relation).

Now with summary sentence S_i and m entities e_{ik} ($k = 1 \ldots m$), $S_i = (wf(e_{i1}), wf(e_{i2}), \ldots, wf(e_{im}))$, where $wf(e_{ik}) = w_k \times f(e_{ik})$, $f(e_{ik})$ is the frequency of e_{ik} and w_k is the weight of e_{ik}. We define $w_k = 1$ if e_{ik} is a common entity and $w_k = 2$ if e_{ik} is a named entity. We give double weight to named entities because of their significance to news articles. After all, a news story typically contains events, places, organizations, people, etc. that denote the news theme. Other things being equal, two sentences sharing a mention of named entities are thematically closer than two sentences sharing a mention of common entities.

Alternatively, we can realize the "concept" as "event" because events are prevalent semantic constructs that bear much of the sentence content in some domains (e.g., narratives and news reports). To represent sentences as event vectors, we can follow Zhang et al.'s (2010) method at the cost of more complexity.

2.2 Sentence Grouping

To meet the global need of identifying sentence groups, we develop two grouping algorithms by applying graph-based operation and clustering.

Connected Component Finding (CC)

This algorithm treats grouping sentences as finding connected components (CC) in a text graph $TG = (V, E)$, where V represents the sentences and E the sentence relations weighted by cosine similarity. Edges with weight $< t$, a threshold, are removed because they represent poor sentence coherence.

The resultant graph may be disconnected, in which we find all of its connected components, using depth-first search. The connected components are the groups we are looking for. Note that this method cannot guarantee that every two sentences in such a group are directly linked, but it does guarantee that there exists a path between every sentence pair.

Modified K-means Clustering (MKM)

Observing that the CC method finds only *coherent groups*, not necessarily *groups of coherent sentences*, we develop a second algorithm using clustering. A good choice might be K-means as it is efficient and outperforms agglomerative clustering methods in NLP applications (Steibach et al., 2000), but the difficulty with the conventional K-means is the decision of K.

Our solution is modified K-means (MKM) based on (Wilpon and Rabiner, 1985). Let's denote

cluster i by CL_i and cluster similarity by $Sim(CL_i) = \underset{S_{im}, S_{in} \in CL_i}{Min}(Sim(S_{im}, S_{in}))$, where $Sim(S_{im}, S_{in})$ is their cosine. The following illustrates the algorithm.

1. CL_1 = all the sentence vectors;
2. Do the 1-means clustering by assigning all the vectors to CL_1;
3. While at least 1 cluster has at least 2 sentences and $Min(Sim(CL_i)) < t$, do:
3.1 If $Sim(S_m, S_n) = Min(Sim(CL_i))$, create two new centroids as S_m and S_n;
3.2 Do the conventional K-means clustering until clusters stabilize;

The above algorithm stops iterating when each cluster contains all above-threshold-similarity sentence pairs or only one sentence. Unlike CC, MKM results in more strongly connected groups, or groups of coherence sentences.

2.3 Ordering Algorithms

After the sentences are grouped, ordering is to be conducted on two levels: group and sentence.

Composed of closely related sentences, groups simulate high-level textual constructs, such as "central event", "cause", "effect", "background", etc. for news articles, around which sentences are generated for global coherence. For an intuitive example, all sentences about "cause" should immediately precede all sentences about "effect" to achieve optimal readability. We propose two approaches to group-level ordering. 1) If the group sentences come from the same document, group (G_i) order is decided by the group-representing sentence (g_i) order ($\prec$ means "precede") in the text.

$$g_i \prec g_j \Rightarrow G_i \prec G_j$$

2) Group order is decided in a greedy fashion in order to maximize the connectedness between adjacent groups, thus enhancing local coherence. Each time a group is selected to achieve maximum similarity with the ordered groups and the first ordered group (G_1) is selected to achieve maximum similarity with all the other groups.

$$G_1 = \arg\max_{G} \sum_{G' \neq G} Sim(G, G')$$

$$G_i = \arg\max_{G \in \{\text{unordered groups}\}} \sum_{j=1}^{i-1} Sim(G_j, G) \quad (i > 1)$$

where $Sim(G, G')$ is the average sentence cosine similarity between G and G'.

Within the ordered groups, sentence-level ordering is aimed to enhance local coherence by placing conceptually close sentences next to each other. Similarly, we propose two approaches. 1) If the sentences come from the same document, they are arranged by the text order. 2) Sentence order is greedily decided. Similar to the decision of group order, with ordered sentence S_{pi} in group G_p:

$$S_{p1} = \arg\max_{S \in G_p} \sum_{S' \neq S} Sim(S, S')$$

$$S_{pi} = \arg\max_{S \in \{\text{unordered sentences in } G_p\}} \sum_{j=1}^{i-1} Sim(S_{pj}, S) \quad (i > 1)$$

Note that the text order is used as a common heuristic, based on the assumption that the sentences are arranged coherently in the source document, locally and globally.

3 Experiments and Preliminary Results

Currently, we have evaluated grouping-based ordering on single-document summarization, for which text order is usually considered sufficient. But there is no theoretical proof that it leads to optimal global and local coherence that concerns us. On some occasions, e.g., a news article adopting the "Wall Street Journal Formula" (Rich and Harper, 2007) where conceptually related sentences are placed at the beginning and the end, sentence conceptual relatedness does not necessarily correlate with spatial proximity and thus selected sentences may need to be rearranged for better readability. We are not aware of any published work that has empirically compared alternative ways of sentence ordering for single-document summarization. The experimental results reported below may draw some attention to this taken-for-granted issue.

3.1 Data and Method

We prepared 3 datasets of 60 documents each, the first (D400) consisting of documents of about 400 words from the Document Understanding Conference (DUC) 01/02 datasets; the second (D1k) consisting of documents of about 1000 words manually selected from popular English journals such as *The Wall Street Journal*, *The Washington Post*, etc; the third (D2k) consisting of documents of about 2000 words from the DUC 01/02 dataset. Then we generated 100-word summaries for D400 and 200-word summaries for D1k and D2k. Since sentence selection is not our

focus, the 180 summaries were all extracts produced by a simple but robust summarizer built on term frequency and sentence position (Aone et al., 1999).

Three human annotators were employed to each provide reference orderings for the 180 summaries and mark paragraph (of at least 2 sentences) boundaries, which will be used by one of the evaluation metrics described below.

In our implementation of the grouping-based ordering, sentences are represented as entity vectors and the threshold $t = Avg(Sim(S_m, S_n)) \times c$, the average sentence similarity in a group multiplied by a coefficient empirically decided on separate held-out datasets of 20 documents for each length category. The "group-representing sentence" is the textually earliest sentence in the group. We experimented with both CC and MKM to generate sentence groups and all the proposed algorithms in 2.3 for group-level and sentence-level orderings, resulting in 8 combinations as test orderings, each coded in the format of "Grouping (CC/MKM) / Group ordering (T/G) / Sentence ordering (T/G)", where T and G represent the text order approach and the greedy selection approach respectively. For example, "CC/T/G" means grouping with CC, group ordering with text order, and sentence ordering with the greedy approach.

We evaluated the test orderings against the 3 reference orderings and compute the average (Madnani et al., 2007) by using 3 different metrics.

The first metric is Kendall's τ (Lapata 2003, 2006), which has been reliably used in ordering evaluations (Bollegala et al., 2006; Madnani et al., 2007). It measures ordering differences in terms of the number of adjacent sentence inversions necessary to convert a test ordering to the reference ordering.

$$\tau = 1 - \frac{4m}{N(N-1)}$$

In this formula, m represents the number of inversions described above and N is the total number of sentences.

The second metric is the Average Continuity (*AC*) proposed by Bollegala et al. (2006), which captures the intuition that the quality of sentence orderings can be estimated by the number of correctly arranged continuous sentences.

$$\mathrm{AC} = exp(1/(k-1)\sum_{n=2}^{k}\log(P_n + \alpha))$$

In this formula, k is the maximum number of continuous sentences, α is a small value in case $P_n = 1$. P_n, the proportion of continuous sentences of length n in an ordering, is defined as $m/(N - n + 1)$ where m is the number of continuous sentences of length n in both the test and reference orderings and N is the total number of sentences. Following (Bollegala et al., 2006), we set $k = Min(4, N)$ and $\alpha = 0.01$.

We also go a step further by considering only the continuous sentences in a paragraph marked by human annotators, because paragraphs are local meaning units perceived by human readers and the order of continuous sentences in a paragraph is more strongly grounded than the order of continuous sentences across paragraph boundaries. So in-paragraph sentence continuity is a better estimation for the quality of sentence orderings. This is our third metric: Paragraph-level Average Continuity (*P-AC*).

$$\text{P-AC} = exp(1/(k-1)\sum_{n=2}^{k}\log(PP_n + \alpha))$$

Here $PP_n = m'/(N - n + 1)$, where m' is the number of continuous sentences of length n in both the test ordering and a paragraph of the reference ordering. All the other parameters are as defined in *AC* and P_n.

3.2 Results

The following tables show the results measured by each metric. For comparison, we also include a "Baseline" that uses the text order. For each dataset, two-tailed t-test is conducted between the top scorer and all the other orderings and statistical significance ($p < 0.05$) is marked with *.

	τ	*AC*	*P-AC*
Baseline	0.6573*	0.4452*	0.0630
CC/T/T	**0.7286**	**0.5688**	**0.0749**
CC/T/G	0.7149	0.5449	0.0714
CC/G/T	0.7094	0.5449	0.0703
CC/G/G	0.6986	0.5320	0.0689
MKM/T/T	0.6735	0.4670*	0.0685
MKM/T/G	0.6722	0.4452*	0.0674
MKM/G/T	0.6710	0.4452*	0.0660
MKM/G/G	0.6588*	0.4683*	0.0682

Table 1: D400 Evaluation

	τ	*AC*	*P-AC*
Baseline	0.3276	0.0867*	0.0428*
CC/T/T	0.3324	0.0979	0.0463*
CC/T/G	0.3276	0.0923	0.0436*
CC/G/T	0.3282	0.0944	0.0479*
CC/G/G	0.3220	0.0893*	0.0428*
MKM/T/T	**0.3390**	**0.1152**	**0.0602**
MKM/T/G	0.3381	0.1130	0.0588
MKM/G/T	0.3375	0.1124	0.0576
MKM/G/G	0.3379	0.1124	0.0581

Table 2: D1k Evaluation

	τ	*AC*	*P-AC*
Baseline	0.3125*	0.1622	0.0213
CC/T/T	**0.3389**	**0.1683**	**0.0235**
CC/T/G	0.3281	**0.1683**	0.0229
CC/G/T	0.3274	0.1665	0.0226
CC/G/G	0.3279	0.1672	0.0226
MKM/T/T	0.3125*	0.1634	0.0216
MKM/T/G	0.3125*	0.1628	0.0215
MKM/G/T	0.3125*	0.1630	0.0216
MKM/G/G	0.3122*	0.1628	0.0215

Table 3: D2k Evaluation

In general, our grouping-based ordering scheme outperforms the baseline for news articles of various lengths and statistically significant improvement can be observed on each dataset. This result casts serious doubt on the widely accepted practice of taking the text order for single-document summary generation, which is a major finding from our study.

The three evaluation metrics give consistent results although they are based on different observations. The P-AC scores are much lower than their AC counterparts because of its strict paragraph constraint.

Interestingly, applying the text order posterior to sentence grouping for group-level and sentence-level ordering leads to consistently optimal performance, as the top scorers on each dataset are almost all "__/T/T". This suggests that the textual realization of coherence can be sought in the source document if possible, after the selected sentences are rearranged. It is in this sense that the general intuition about the text order is justified. It also suggests that tightly knit paragraphs (groups), where the sentences are closely connected, play a crucial role in creating a coherence flow. Shuffling those paragraphs may not affect the final coherence[1].

The grouping method does make a difference. While CC works best for the short and long datasets (D400 and D2k), MKM is more effective for the medium-sized dataset D1k. Whether the difference is simply due to length or linguistic/stylistic subtleties is an interesting topic for in-depth study.

4 Conclusion and Future Work

We have established a grouping-based ordering scheme to accommodate both local and global coherence for summary generation. Experiments on single-document summaries validate our approach and challenge the well accepted text order by the summarization community.

Nonetheless, the results do not necessarily propagate to multi-document summarization, for which the same-document clue for ordering cannot apply directly. Adapting the proposed scheme to multi-document summary generation is the ongoing work we are engaged in. In the next step, we will experiment with alternative sentence representations and ordering algorithms to achieve better performance.

We are also considering adapting more sophisticated coherence-oriented models, such as (Soricut and Marcu, 2006; Elsner et al., 2007), to our problem so as to make more interesting comparisons possible.

Acknowledgements

The reported work was inspired by many talks with my supervisor, Dr. Wenjie Li, who saw through this work down to every writing detail. The author is also grateful to many people for assistance. You Ouyang shared part of his summarization work and helped with the DUC data. Dr. Li Shen, Dr. Naishi Liu, and three participants helped with the experiments. I thank them all.

The work described in this paper was partially supported by Hong Kong RGC Projects (No. PolyU 5217/07E).

[1] I thank an anonymous reviewer for pointing this out.

References

Aone, C., Okurowski, M. E., Gorlinsky, J., and Larsen, B. 1999. A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. 71–80. Cambridge, Massachusetts: MIT Press.

Barzilay, R., Elhadad, N., and McKeown, K. 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research*, 17: 35–55.

Barzilay, R. and Lapata, M. 2005. Modeling Local Coherence: An Entity-based Approach. In *Proceedings of the 43rd Annual Meeting of the ACL*, 141–148. Ann Arbor.

Barzilay, R. and Lapata, M. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34: 1–34.

Barzilay, R. and Lee L. 2004. Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization. In *HLT-NAACL 2004: Proceedings of the Main Conference*. 113–120.

Bollegala, D, Okazaki, N., and Ishizuka, M. 2006. A Bottom-up Approach to Sentence Ordering for Multi-document Summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, 385–392. Sydney.

Elsner, M., Austerweil, j. & Charniak E. 2007. "A Unified Local and Global Model for Discourse Coherence". In *Proceedings of NAACL HLT 2007*, 436-443. Rochester, NY.

Grosz, B. J., Aravind K. J., and Scott W. 1995. Centering: A framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–225.

Halliday, M. A. K., and Hasan, R. 1976. *Cohesion in English*. London: Longman.

Hovy, E. 2005. Automated Text Summarization. In R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, pp. 583–598. Oxford: Oxford University Press.

Jing, H. 2000. Sentence Reduction for Automatic Text Summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, WA, pp. 310–315.

Jing, H., and McKeown, K. 2000. Cut and Paste Based Text Summarization. In *Proceedings of the 1st NAACL*, 178–185.

Kehler, A. 2002. *Coherence, Reference, and the Theory of Grammar*. Stanford, California: CSLI Publications.

Lapata, M. 2003. Probabilistic Text Structuring: Experiments with Sentence Ordering. In *Proceedings of the Annual Meeting of ACL*, 545–552. Sapporo, Japan.

Lapata, M. 2006. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):1–14.

Madnani, N., Passonneau, R., Ayan, N. F., Conroy, J. M., Dorr, B. J., Klavans, J. L., O'leary, D. P., and Schlesinger, J. D. 2007. Measuring Variability in Sentence Ordering for News Summarization. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, 81–88. Germany.

Mann, W. C. and Thompson, S. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8:243–281.

Rich C., and Harper, C. 2007. *Writing and Reporting News: A Coaching Method, Fifth Edition*. Thomason Learning, Inc. Belmont, CA.

Soricut, R. and Marcu D. 2006. Discourse Generation Using Utility-Trained Coherence Models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 803–810.

Steibach, M., Karypis, G., and Kumar V. 2000. A Comparison of Document Clustering Techniques. Technical Report 00-034. Department of Computer Science and Engineering, University of Minnesota.

Tapiero, I. 2007. *Situation Models and Levels of Coherence: Towards a Definition of Comprehension*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Wilpon, J. G. and Rabiner, L. R. 1985. A Modified K-means Clustering Algorithm for Use in Isolated Word Recognition. In *IEEE Trans. Acoustics, Speech, Signal Proc*. ASSP-33(3), 587–594.

Zhang R., Li, W., and Lu, Q. 2010. Sentence Ordering with Event-Enriched Semantics and Two-Layered Clustering for Multi-Document News Summarization. In *COLING 2010: Poster Volume*, 1489–1497, Beijing.

Pre- and Postprocessing for Statistical Machine Translation into Germanic Languages

Sara Stymne
Department of Computer and Information Science
Linköping University, Linköping, Sweden
sara.stymne@liu.se

Abstract

In this thesis proposal I present my thesis work, about pre- and postprocessing for statistical machine translation, mainly into Germanic languages. I focus my work on four areas: compounding, definite noun phrases, reordering, and error correction. Initial results are positive within all four areas, and there are promising possibilities for extending these approaches. In addition I also focus on methods for performing thorough error analysis of machine translation output, which can both motivate and evaluate the studies performed.

1 Introduction

Statistical machine translation (SMT) is based on training statistical models from large corpora of human translations. It has the advantage that it is very fast to train, if there are available corpora, compared to rule-based systems, and SMT systems are often relatively good at lexical disambiguation. A large drawback of SMT systems is that they use no or little grammatical knowledge, relying mainly on a target language model for producing correct target language texts, often resulting in ungrammatical output. Thus, methods to include some, possibly shallow, linguistic knowledge seem reasonable.

The main focus for SMT to date has been on translation into English, for which the models work relatively well, especially for source languages that are structurally similar to English. There has been less research on translation out of English, or between other language pairs. Methods that are useful for translation into English have problems in many cases, for instance for translation into morphologically rich languages. Word order differences and morphological complexity of a language have been shown to be explanatory variables for the performance of phrase-based SMT systems (Birch et al., 2008). German and the Scandinavian languages are a good sample of languages, I believe, since they are both more morphologically complex than English to a varying degree, and the word order differ to some extent, with mostly local differences between English and Scandinavian, and also long distance differences with German, especially for verbs.

Some problems with SMT into German and Swedish are exemplified in Table 1. In the German example, the translation of the verb *welcome* is missing in the SMT output. Missing and misplaced verbs are common error types, since the German verb should appear last in the sentence in this context, as in the reference, *begrüßen*. There is also an idiomatic compound, *redebeitrag* (*speech+contribution; intervention*) in the reference, which is produced as the single word *beitrag* in the SMT output. In the Swedish example, there are problems with a definite NP, which has the wrong gender of the definite article, *den* instead of *det*, and is missing a definite suffix on the noun *synsätt(et)* (*(the) approach*).

In this proposal I outline my thesis work which aims to improve statistical machine translation, particularly into Germanic languages, by using pre- and postprocessing on one or both language sides, with an additional focus on error analysis. In section 2 I present a thesis overview, and in section 3 I briefly overview MT evaluation techniques, and discuss my work on MT error analysis. In section 4 I describe my work on pre- and postprocessing, which is focused on compounding, definite noun phrases, word order, and error correction.

Proceedings of the ACL-HLT 2011 Student Session, pages 12–17,
Portland, OR, USA 19-24 June 2011.

En source	I too would like to welcome Mr Prodi's forceful and meaningful intervention.
De SMT	Ich möchte auch herrn Prodis energisch und sinnvollen Beitrag.
De reference	Ich möchte meinerseits auch den klaren und substanziellen Redebeitrag von Präsident Prodi begrüßen.
En source	So much for the scientific approach.
Se SMT	Så mycket för den vetenskapliga synsätt.
Se reference	Så mycket för den vetenskapliga infallsvinkeln.

Table 1: Examples of problematic PBSMT output

2 Thesis Overview

My main research focus is how pre- and postprocessing can be used to improve statistical MT, with a focus on translation into Germanic languages. The idea behind preprocessing is to change the training corpus on the source side and/or on the target side in order to make them more similar, which makes the SMT task easier, since the standard SMT models work better for more similar languages. Postprocessing is needed after the translation when the target language has been preprocessed, in order to restore it to the normal target language. Postprocessing can also be used on standard MT output, in order to correct some of the errors from the MT system. I focus my work about pre- and postprocessing on four areas: compounding, definite noun phrases, word order, and error correction. In addition I am making an effort into error analysis, to identify and classify errors in the MT output, both in order to focus my research effort, and to evaluate and compare systems.

My work is based on the phrase-based approach to statistical machine translation (PBSMT, Koehn et al. (2003)). I further use the framework of factored machine translation, where each word is represented as a vector of factors, such as surface word, lemma and part-of-speech, rather than only as surface words (Koehn and Hoang, 2007). I mostly utilize factors to translate into both words and (morphological) part-of-speech, and can then use an additional sequence model based on part-of-speech, which potentially can improve word order and agreement. I take advantage of available tools, such as the Moses toolkit (Koehn et al., 2007) for factored phrase-based translation.

I have chosen to focus on PBSMT, which is a very successful MT approach, and have received much research focus. Other SMT approaches, such as hierarchical and syntactical SMT (e.g. Chiang (2007), Zhang et al. (2007a)) can potentially overcome some language differences that are problematic for PBSMT, such as long-distance word order differences. Many of these models have had good results, but they have the drawback of being more complex than PBSMT, and some methods do not scale well to large corpora. While these models at least in principle address some of the drawbacks of the flat structure in PBSMT, Wang et al. (2010) showed that a syntactic SMT system can still gain from preprocessing such as parse-tree modification.

3 Evaluation and Error Analysis

Machine translation systems are often only evaluated quantitatively by using automatic metrics, such as Bleu (Papineni et al., 2002), which compares the system output to one or more human reference translations. While this type of evaluation has its advantages, mainly that it is fast and cheap, its correlation with human judgments is often low, especially for translation out of English (Callison-Burch et al., 2009). In order to overcome these problems to some extent I use several metrics in my studies, instead of only Bleu. Despite this, metrics only give a single score per sentence batch and system, which even using several metrics gives us little information on the particular problems with a system, or about what the possible improvements are.

One alternative to automatic metrics is human judgments, either absolute scores, for instance for adequacy or fluency, or by ranking sentences or segments. Such evaluations are a valuable complement to automatic metrics, but they are costly and time-consuming, and while they are useful for comparing systems they also fail to pinpoint specific problems. I mainly take advantage of this type of evaluation as part of participating with my research group in MT

shared tasks with large evaluation campaigns such as WMT (e.g. Callison-Burch et al. (2009)).

To overcome the limitation of quantitative evaluations, I focus on error analysis (EA) of MT output in my thesis. EA is the task of annotating and classifying the errors in MT output, which gives a qualitative view. It can be used to evaluate and compare systems, but is also useful in order to focus the research effort on common problems for the language pair in question. There have been previous attempts of describing typologies for EA for MT, but they are not unproblematic. Vilar et al. (2006) suggested a typology with five main categories: missing, incorrect, unknown, word order, and punctuation, which have also been used by other researchers, mainly for evaluation. However, this typology is relatively shallow and mixes classification of errors with causes of errors. Farrús et al. (2010) suggested a typology based on linguistic categories, such as orthography and semantics, but their descriptions of these categories and their subcategories are not detailed. Thus, as part of my research, I am in the progress of designing a fine-grained typology and guidelines for EA. I have also created a tool for performing MT error analysis (Stymne, 2011a). Initial annotations have helped to focus my research efforts, and will be discussed below. I also plan to use EA as one means of evaluating my work on pre- and postprocessing.

4 Main Research Problems

In this section I describe the four main problem areas I will focus on in my thesis project. I summarize briefly previous work in each area, and outline my own current and planned contributions. Sample results from the different studies are shown in Table 2.

4.1 Compounding

In most Germanic languages, compounds are written without spaces or other word boundaries, which makes them problematic for SMT, mainly due to sparse data problems. The standard method for treating compounds for translation from Germanic languages is to split them in both the training data and translation input (e.g. (Nießen and Ney, 2000; Koehn and Knight, 2003; Popović et al., 2006)). Koehn and Knight (2003) also suggested a corpus-based compound splitting method that has been much used for SMT, where compounds are split based on corpus frequencies of its parts.

If compounds are split for translation into Germanic languages, the SMT system produces output with split compounds, which need to be postprocessed into full compounds. There has been very little research into this problem. For this process to be successful, it is important that the SMT system produces the split compound parts in a correct word order. To encourage this I have used a factored translation system that outputs parts-of-speech and uses a sequence model on parts-of-speech. I extended the part-of-speech tagset to use special part-of-speech tags for split compound parts, which depend on the head part-of-speech of the compound. For instance, the Swedish noun *päronträd* (*pear tree*) would be tagged as *päron|N-part träd|N* when split. Using this model the number of compound parts that were produced in the wrong order was reduced drastically compared to not using a part-of-speech sequence model for translation into German (Stymne, 2009a).

I also designed an algorithm for the merging task that uses these part-of-speech tags to merge compounds only when the next part-of-speech tag matches. This merging method outperforms reimplementations and variations of previous merging suggestions (Popović et al., 2006), and methods adapted from morphology merging (Virpioja et al., 2007) for translation into German (Stymne, 2009a). It also has the advantage over previous merging methods that it can produce novel compounds, while at the same time reducing the risk of merging parts into non-words. I have also shown that these compound processing methods work equally well for translation into Swedish (Stymne and Holmqvist, 2008). Currently I am working on methods for further improving compound merging, with promising initial results.

4.2 Definite Noun Phrases

In Scandinavian languages there are two ways to express definiteness in noun phrases, either by a definite article, or by a suffix on the noun. This leads to problems when translating into these languages, such as superfluous definite articles and wrong forms of nouns. I am not aware of any published research in this area, but an unpublished

Language pair	Corpus	Corpus size	Testset size	In article	System	Bleu	NIST
En-De	Europarl	439,513	2,000	Stymne (2008)	BL	19.31	5.727
					+Comp	19.73	5.854
En-Se	Europarl	701,157	2,000	Stymne and Holmqvist (2008)	BL	21.63	6.109
					+Comp	22.12	6.143
En-Da	Automotive	168,046	1,000	Stymne (2009b)	BL	70.91	8.816
					+Def	76.35	9.363
En-Se	Europarl	701,157	1,000	Stymne (2011b)	BL	21.63	6.109
					+Def	22.03	6.178
En-De	Europarl	439,513	2,000	Stymne (2011c)	BL	19.32	5.901
					+Reo	19.59	5.936
En-Se	Europarl	701,157	335	Stymne and Ahrenberg (2010)	BL	19.44	5.381
					+EC	22.12	5.447

Table 2: A selection of results for the four pre- and postprocessing strategies. Corpus sizes are given as number of sentences. BL is baseline systems, +Comp with compound processing, +Def with definite processing, +Reo with iterative reordering and alignment and monotone decoding, +EC with grammar checker error correction. The test set for error correction only contains sentences that are affected by the error correction.

report shows no gain for a simple pre-processing strategy for translation from German to Swedish (Samuelsson, 2006). There is similar work on other phenomena, such as Nießen and Ney (2000), who move German separated verb prefixes, to imitate the English phrasal verb structure.

I address definiteness by preprocessing the source language, to make definite NPs structurally similar to target language NPs. The transformations are rule-based, using part-of-speech tags. Definite NPs in Scandinavian languages are mimicked in the source language by removing superfluous definite articles, and/or adding definite suffixes to nouns. In an initial study, this gave very good results, with relative Bleu improvements of up to 22.1% for translation into Danish (Stymne, 2009b). In Swedish and Norwegian, the distribution of definite suffixes is more complex than in Danish, and the basic strategy that worked well for Danish was not successful (Stymne, 2011b). A small modification to the basic strategy, so that superfluous English articles were removed, but no suffixes were added, was successful for translation from English into Swedish and Norwegian. A planned extension is to integrate the transformations into a lattice that is fed to the decoder, in the spirit of (Dyer et al., 2008).

4.3 Word Order

There has been a lot of research on how to handle word order differences between languages. Preprocessing approaches can use either hand-written rules targeting known language differences (e.g. Collins et al. (2005), Li et al. (2009)), or automatically learnt rules (e.g. Xia and McCord (2004), Zhang et al. (2007b)), which are basically language independent.

I have performed an initial study on a language independent word order strategy where reordering rule learning and word alignment are performed iteratively, since they both depend on the other process (Stymne, 2011c). There were no overall improvements as measured by Bleu, but an investigation of the reordering rules showed that the rules learned in the different iterations are different with regard to the linguistic phenomena they handle, indicating that it is possible to learn new information from iterating rule learning and word alignment. In this study I only choose the 1-best reordering as input to the SMT system. I plan to extend this by presenting several reorderings to the decoder as a lattice, which has been successful in previous work (see e.g. Zhang et al. (2007b)).

My preliminary error analysis has shown that there are two main word order difficulties for translation between English and Swedish, adverb placement, and V2 errors, where the verb is not placed in the correct position when it should be placed before the subject. I plan to design a preprocessing scheme to tackle these particular problems for English-Swedish translation.

4.4 Error Correction

Postprocessing can be used to correct MT output that has not been preprocessed, for instance in order to improve the grammaticality. There has not been much research in this area. A few examples are Elming (2006), who use transformation-based learning for word substitution based on aligned human post-edited sentences, and Guzmán (2007) who used regular expression to correct regular Spanish errors. I have applied error correction suggestions given by a grammar checker to the MT output, showing that it can improve certain types of errors, such as NP agreement and word order, with a high precision, but unfortunately with a low recall (Stymne and Ahrenberg, 2010). Since the recall is low, the positive effect on metrics such as Bleu is small on general test sets, but there are improvements on test sets which only contains sentences that are affected by the postprocessing. An error analysis showed that 68–74% of the corrections made were useful, and only around 10% of the changes made were harmful. I believe that this approach could be even more useful for similar languages, such as Danish and Swedish, where a spell-checker might also be useful.

The initial error analysis I have performed has helped to identify common errors in SMT output, and shown that many of them are quite regular. A strategy I intend to pursue is to further identify common and regular problems, and to either construct rules or to train a machine learning classifier to identify them, in order to be able to postprocess them. It might also be possible to use the annotations from the error analysis as part of the training data for such a classifier.

5 Discussion

The main focus of my thesis will be on designing and evaluating methods for pre- and postprocessing of statistical MT, where I will contribute methods that can improve translation within the four areas discussed in section 4. The effort is focused on translation into Germanic languages, including German, on which there has been much previous research, and Swedish and other Scandinavian languages, where there has been little previous research. I believe that both language-pair dependent and independent methods for pre- and postprocessing can be useful. It is also the case that some language-pair dependent methods carry over to other (similar) language pairs with no or little modification. So far I have mostly used rule-based processing, but I plan to extend this with investigating machine learning methods, and compare the two main approaches.

I strongly believe that it is important for MT researchers to perform qualitative evaluations, both for identifying problems with MT systems, and for evaluating and comparing systems. In my experience it is often the case that a change to the system to improve one aspect, such as compounding, also leads to many other changes, in the case of compounding for instance because of the possibility of improved alignments, which I think we lack a proper understanding of.

My planned thesis contributions are to design a detailed error typology, guidelines, and a tool, targeted at MT researchers, for performing error annotation, and to improve statistical machine translation in four problem areas, using several methods of pre- and postprocessing.

References

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of EMNLP*, pages 745–754, Honolulu, Hawaii, USA.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of WMT*, pages 1–28, Athens, Greece.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):202–228.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL*, pages 531–540, Ann Arbor, Michigan, USA.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL*, pages 1012–1020, Columbus, Ohio, USA.

Jakob Elming. 2006. Transformation-based correction of rule-based MT. In *Proceedings of EAMT*, pages 219–226, Oslo, Norway.

Mireia Farrús, Marta R. Costa-jussà, José B. Mariño, and José A. R. Fonollosa. 2010. Linguistic-based evaluation criteria to identify statistical machine translation

errors. In *Proceedings of EAMT*, pages 52–57, Saint Raphaël, France.

Rafael Guzmán. 2007. Advanced automatic MT post-editing using regular expressions. *Multilingual*, 18(6):49–52.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proceedings of EMNLP/CoNLL*, pages 868–876, Prague, Czech Republic.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of EACL*, pages 187–193, Budapest, Hungary.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54, Edmonton, Alberta, Canada.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL, demonstration session*, pages 177–180, Prague, Czech Republic.

Jin-Ji Li, Jungi Kim, Dong-Il Kim, and Jong-Hyeok Lee. 2009. Chinese syntactic reordering for adequate generation of Korean verbal phrases in Chinese-to-Korean SMT. In *Proceedings of WMT*, pages 190–196, Athens, Greece.

Sonja Nießen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *Proceedings of CoLing*, pages 1081–1085, Saarbrücken, Germany.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, Pennsylvania, USA.

Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical machine translation of German compound words. In *Proceedings of FinTAL – 5th International Conference on Natural Language Processing*, pages 616–624, Turku, Finland. Springer Verlag, LNCS.

Yvonne Samuelsson. 2006. Nouns in statistical machine translation. Unpublished manuscript: Term paper, Statistical Machine Translation.

Sara Stymne and Lars Ahrenberg. 2010. Using a grammar checker for evaluation and postprocessing of statistical machine translation. In *Proceedings of LREC*, pages 2175–2181, Valetta, Malta.

Sara Stymne and Maria Holmqvist. 2008. Processing of Swedish compounds for phrase-based statistical machine translation. In *Proceedings of EAMT*, pages 180–189, Hamburg, Germany.

Sara Stymne. 2008. German compounds in factored statistical machine translation. In *Proceedings of GoTAL – 6th International Conference on Natural Language Processing*, pages 464–475, Gothenburg, Sweden. Springer Verlag, LNCS/LNAI.

Sara Stymne. 2009a. A comparison of merging strategies for translation of German compounds. In *Proceedings of EACL, Student Research Workshop*, pages 61–69, Athens, Greece.

Sara Stymne. 2009b. Definite noun phrases in statistical machine translation into Danish. In *Proceedings of the Workshop on Extracting and Using Constructions in NLP*, pages 4–9, Odense, Denmark.

Sara Stymne. 2011a. Blast: A tool for error analysis of machine translation output. In *Proceedings of ACL, demonstration session*, Portland, Oregon, USA.

Sara Stymne. 2011b. Definite noun phrases in statistical machine translation into Scandinavian languages. In *Proceedings of EAMT*, Leuven, Belgium.

Sara Stymne. 2011c. Iterative reordering and word alignment for statistical MT. In *Proceedings of the 18th Nordic Conference on Computational Linguistics*, Riga, Latvia.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error analysis of machine translation output. In *Proceedings of LREC*, pages 697–702, Genoa, Italy.

Sami Virpioja, Jaako J. Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *Proceedings of MT Summit XI*, pages 491–498, Copenhagen, Denmark.

Wei Wang, Jonathan May, Kevin Knight, and Daniel Marcu. 2010. Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Computational Linguistics*, 36(2):247–277.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of CoLing*, pages 508–514, Geneva, Switzerland.

Min Zhang, Hongfei Jiang, Ai Ti Aw, Jun Sun, Sheng Li, and Chew Lim Tan. 2007a. A tree-to-tree alignment-based model for statistical machine translation. In *Proceedings of MT Summit XI*, pages 535–542, Copenhagen, Denmark.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007b. Improved chunk-level reordering for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 21–28, Trento, Italy.

Exploring Entity Relations for Named Entity Disambiguation

Danuta Ploch
DAI-Labor, Technische Universität Berlin
Berlin, Germany
danuta.ploch@dai-labor.de

Abstract

Named entity disambiguation is the task of linking an entity mention in a text to the correct real-world referent predefined in a knowledge base, and is a crucial subtask in many areas like information retrieval or topic detection and tracking. Named entity disambiguation is challenging because entity mentions can be ambiguous and an entity can be referenced by different surface forms. We present an approach that exploits Wikipedia relations between entities co-occurring with the ambiguous form to derive a range of novel features for classifying candidate referents. We find that our features improve disambiguation results significantly over a strong popularity baseline, and are especially suitable for recognizing entities not contained in the knowledge base. Our system achieves state-of-the-art results on the TAC-KBP 2009 dataset.

1 Introduction

Identifying the correct real-world referents of named entities (NE) mentioned in text (such as people, organizations, and geographic locations) plays an important role in various natural language processing and information retrieval tasks. The goal of Named Entity Disambiguation (NED) is to label a surface form denoting an NE in text with one of multiple predefined NEs from a knowledge base (KB), or to detect that the surface form refers to an out-of-KB entity, which is known as NIL detection. NED has become a popular research field recently, as the growth of large-scale publicly available encyclopedic knowledge resources such as Wikipedia has stimulated research on linking NEs in text to their entries in these KBs (Bunescu and Pasca, 2006; McNamee and Dang, 2009).

The disambiguation of named entities raises several challenges: Surface forms in text can be ambiguous, and the same entity can be referred to by different surface forms. For example, the surface form "George Bush" may denote either of two former U.S. presidents, and the later president can be referred to by "George W. Bush" or with his nickname "Dubya". Thus, a many-to-many mapping between surface forms and entities has to be resolved. In addition, entity mentions may not have a matching entity in the KB, which is often the case for non-popular entities.

Typical approaches to NED combine the use of document context knowledge with entity information stored in the KB in order to disambiguate entities. Many systems represent document context and KB information as word or concept vectors, and rank entities using vector space similarity metrics (Cucerzan, 2007). Other authors employ supervised machine learning algorithms to classify or rank candidate entities (Bunescu and Pasca, 2006; Zhang et al., 2010). Common features include popularity metrics based on Wikipedia's graph structure or on name mention frequency (Dredze et al., 2010; Han and Zhao, 2009), similarity metrics exploring Wikipedia's concept relations (Han and Zhao, 2009), and string similarity features. Recent work also addresses the task of NIL detection (Dredze et al., 2010).

While previous research has largely focused on disambiguating each entity mention in a document

Proceedings of the ACL-HLT 2011 Student Session, pages 18–23,
Portland, OR, USA 19-24 June 2011.

separately (McNamee and Dang, 2009), we explore an approach that is driven by the observation that entities normally co-occur in texts. Documents often discuss several different entities related to each other, e.g. a news article may report on a meeting of political leaders from different countries. Analogously, entries in a KB such as Wikipedia are linked to other, related entries.

Our Contributions In this paper, we evaluate a range of novel disambiguation features that exploit the relations between NEs identified in a document and in the KB. Our goal is to explore the usefulness of Wikipedia's link structure as source of relations between entities. We propose a method for candidate selection that is based on an inverted index of surface forms and entities (Section 3.2). Instead of a bag-of-words approach we use co-occurring NEs in text for describing an ambiguous surface form. We introduce several different disambiguation features that exploit the relations between entities derived from the graph structure of Wikipedia (Section 3.3). Finally, we combine our disambiguation features and achieve state-of-the-art results with a Support Vector Machine (SVM) classifier (Section 4).

2 Problem statement

The task of NED is to assign a surface form s found in a document d to a target NE $t \in E(s)$, where $E(s) \subset E$ is a set of candidate NEs from an entity KB that is defined by $E = \{e_1, e_2, ..., e_n\}$, or to recognize that the found surface form s refers to a missing target entity $t \notin E(s)$. For solving the task, three main challenges have to be addressed:

Ambiguity Names of NEs may be ambiguous. Since the same surface form s may refer to more than one NE e, the correct target entity t has to be determined from a set of candidates $E(s)$

Name variants Often, name variants (e.g. abbreviations, acronyms or synonyms) are used in texts to refer to the same NE, which has to be considered for the determination of candidates $E(s)$ for a given surface form s.

KB coverage KBs cover only a limited number of NEs, mostly popular NEs. Another challenge of NED is therefore to recognize missing NEs where $t \notin E(s)$, given a surface form s (NIL detection).

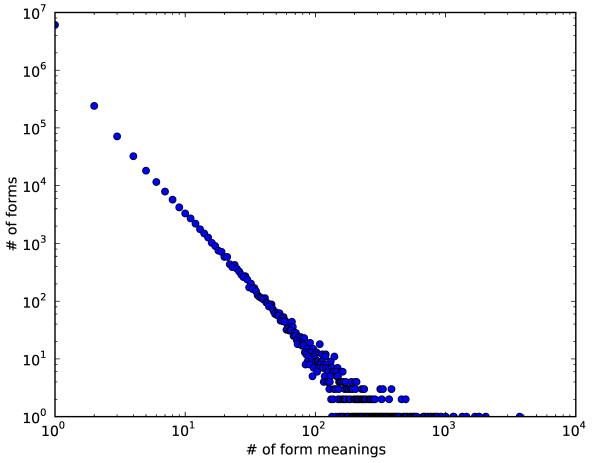

Figure 1: Ambiguity of Wikipedia surface forms. The distribution follows a power law, as many surface forms have only a single meaning (i.e. refer to a single Wikipedia concept), and some surface forms are highly ambiguous, referring to very many different concepts.

3 Named Entity Disambiguation

We formulate NED as a supervised binary classification problem. In this section we describe the construction and structure of the KB and the candidate selection scheme, followed by an overview of disambiguation features and the candidate classification algorithm.

3.1 Knowledge base construction

Our approach disambiguates named entities against a KB constructed from Wikipedia. To this end, we process Wikipedia to extract several types of information for each Wikipedia article describing a concept (i.e. any article not being a redirect page, a disambiguation page, or any other kind of meta page). We collect a set of name variants (surface forms) for each concept from article titles, redirect pages, disambiguation pages and the anchor texts of internal Wikipedia links, following Cucerzan (2007). For each concept, we also collect its set of incoming and outgoing links to other Wikipedia pages. Finally, we extract the article's full text. We store this information in an inverted index, which allows for very efficient access and search during candidate selection and feature computation.

The distribution of surface forms follows a power law, where the majority of surface forms is unambiguous, but some surface forms are very ambiguous (Figure 1). This suggests that for a given set of distinct surface forms found in a document, many of these will unambiguously refer to a single Wikipedia entity. These entities can then be used to disambiguate surface forms referring to multiple entities.

3.2 Candidate selection

Given a surface form identified in a document, the task of the candidate selection component is to retrieve a set of candidate entities from the KB. To this end, we execute a search on index fields storing article titles, redirect titles, and name variants. We implement a weighted search to give high weights to exact title matches, a lesser emphasis on redirect matches, and finally a low weight for all other name variants. In addition, we implement a fuzzy search on the title and redirect fields to select KB entries with approximate string similarity to the surface form.

3.3 Disambiguation features

In this section, we describe the features that we use in our disambiguation approach.

Entity Context (EC) The EC disambiguation feature is calculated as the cosine similarity between the document context **d** of a surface form s and the Wikipedia article **c** of each candidate $c \in E(s)$. We represent both contexts as vectors of URIs. To create **d** we extract all NEs from the text using the Stanford NE Recognizer (Finkel et al., 2005) and represent each NE by its Wikipedia URI. If a surface form is ambiguous, we choose the most popular NE with the popularity metric described below. Analogously, we represent each c as a vector of the incoming and outgoing URIs found on its Wikipedia page.

Link Context (LC) The link context feature is an extension of the EC feature. Since our observations have shown that the entity context can be very small and consequently the overlap between **d** and **c** may be very low, we extend **d** by all incoming (LC-in) or by all incoming and outgoing (LC-all) Wikipedia URIs of the NEs from the entity context. We assume that Wikipedia pages that refer to other Wikipedia pages contain information on the referenced pages or at least are thematically related to these pages. With the extension of **d** to **d'**, we expect a higher overlap between the context vectors, so that $\cos(\mathbf{d'}, \mathbf{c}) \geq \cos(\mathbf{d}, \mathbf{c})$.

Candidate Rank (CR) The features described so far disambiguate every surface form $s \in S$ from a document d separately, whereas our Candidate Rank feature aims to disambiguate all surface forms S found in a document d at once. We represent d as a graph $D = (E(S), L(E(S)))$ where the nodes $E(S) = \cup_{s \in S} E(s)$ are all candidates of all surface forms in the document and $L(E(S))$ is the set of links between the candidates, as found in Wikipedia. Then, we compute the PageRank score (Brin and Page, 1998) of all $c \in E(S)$ and choose for each s the candidate with the highest PageRank score in the document graph D.

Standard Features In addition to the previously described features we also implement a set of commonly accepted features. These include a feature based on the cosine similarity between word vector representations of the document and the Wikipedia article of each candidate (BOW) (Bunescu, 2007). We perform stemming, remove stopwords, and weight words with tf.idf in both cases. Another standard feature we use is the popularity of a surface form (SFP). We calculate how often a surface form s references a candidate $c \in E(s)$ in relation to the total number of mentions of s in Wikipedia (Han and Zhao, 2009). Since we use an index for selecting candidates (Section 3.2), we also exploit the candidate selection score (CS) returned for each candidate as a disambiguation feature.

3.4 Candidate classifier and NIL detection

We cast NED as a supervised classification task and use two binary SVM classifiers (Vapnik, 1995). The first classifier decides for each candidate $c \in E(s)$ if it corresponds to the target entity. Each candidate is represented as a vector $\mathbf{x}^{(c)}$ of features. For training the classifier we label as a positive example at most one $\mathbf{x}^{(c)}$ from the set of candidates for a surface form s, and all others as negative.

In addition, we train a separate classifier to detect NIL queries, i.e. where all $\mathbf{x}^{(c)}$ from $E(s)$ are labeled as negative examples. This may e.g. be the case

	All queries	KB	NIL
Baseline features	0.7797	0.6246	0.8964
All features	0.8391	0.6795	0.9592
Best features	0.8422	0.6825	0.9623
Dredze et al.	0.7941	0.6639	0.8919
Zheng et al.	0.8494	0.7900	0.8941
Best TAC 2009	0.8217	0.7725	0.8919
Median TAC 2009	0.7108	0.6352	0.7891

Table 1: Micro-averaged accuracy for TAC-KBP 2009 data compared for different feature sets. The best feature set contains all features except for LC-all and CR. Our system outperforms previously reported results on NIL queries, and compares favorably on all queries.

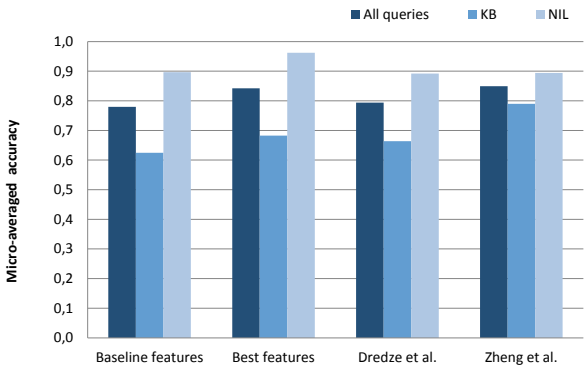

Figure 2: The micro-averaged accuracy for all types of queries on TAC-KBP 2009 data in comparison to other systems.

if the similarity values of all candidates $c \in E(s)$ are very low. We calculate several different features, such as the maximum, mean and minimum, the difference between maximum and mean, and the difference between maximum and minimum, of all atomic features, using the feature vectors of all candidates in $E(s)$. Both classifier use a radial basis function kernel, with parameter settings of $C = 32$ and $\gamma = 8$. We optimized these settings on a separate development dataset.

4 Evaluation

We conduct our experiments on the 2009 Knowledge Base Population (KBP) dataset of the Text Analysis Conference (TAC) (McNamee and Dang, 2009). The dataset consists of a KB derived from a 2008 snapshot of the English Wikipedia, and a collection of newswire, weblog and newsgroup documents. A set of 3904 surface form-document pairs (queries) is constructed from these sources, encompassing 560 unique entities. The majority of queries (57%) are NIL queries, of the KB queries, 69% are for organizations and 15% each for persons and geopolitical entities. For each query the surface form appearing in the given document has to be disambiguated against the KB.

We randomly split the 3904 queries to perform 10-fold cross-validation, and stratify the resulting folds to ensure a similar distribution of KB and NIL queries in our training data. After normalizing feature values to be in $[0, 1]$, we train a candidate and a NIL classifier on 90% of the queries in each iteration, and test using the remaining 10%. Results reported in this paper are then averaged across the test folds.

Table 1 compares the micro-averaged accuracy of our approach on KB and NIL queries for different feature sets, and lists the results of two other state-of-the-art systems (Dredze et al., 2010; Zheng et al., 2010), as well as the best and median reported performance of the 2009 TAC-KBP track (McNamee et al., 2010). Micro-averaged accuracy is calculated as the fraction of correct queries, and is the official TAC-KBP evaluation measure. As a baseline we use a feature set consisting of the BOW and SFP features. The best feature set in our experiments comprises all features except for the LC-all and CR features.

Our best accuracy of 0.84 compares favorably with other state-of-the-art systems on this dataset. Using the best feature set improves the disambiguation accuracy by 6.2% over the baseline feature set, which is significant at $p = 0.05$. For KB queries our system's accuracy is higher than that of Dredze et al., but lower than the accuracy reported by Zheng et al. One striking result is the high accuracy for NIL queries, where our approach outperforms all previously reported results (Figure 2).

Figure 3 displays the performance of our approach when iteratively adding features. We can see that the novel entity features contribute to a higher overall accuracy. Including the candidate selection score (CS) improves accuracy by 3.6% over the baseline. The Wikipedia link-based features provide additional gains, however differences are quite

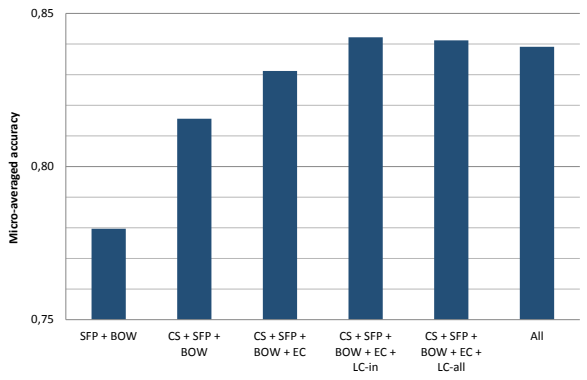

Figure 3: Differences in micro-averaged accuracy for various feature combinations on TAC-KBP 2009 data. Adding Wikipedia link-based features significantly improves performance over the baseline feature set.

small ($1.0 - 1.5\%$). We find that there is hardly any difference in performance between using the LC-all and LC-in features. The Candidate Rank (CR) feature slightly decreases the overall accuracy. A manual inspection of the CR feature shows that often candidates cannot be distinguished by the classifier because they are assigned the same PageRank scores. We assume this results from our use of uniform priors for the edges and vertices of the document graphs.

5 Conclusion and Future Work

We presented a supervised approach for named entity disambiguation that explores novel features based on Wikipedia's link structure. These features use NEs co-occurring with an ambiguous surface form in a document and their Wikipedia relations to score the candidates. Our system achieves state-of-the-art results on the TAC-KBP 2009 dataset. We find that our features improve disambiguation results by 6.2% over the popularity baseline, and are especially helpful for recognizing entities not contained in the KB.

In future work we plan to explore multilingual data for NED. Since non-English versions of Wikipedia often are less extensive than the English version we find it promising to combine Wikipedia versions of different languages and to use them as a source for multilingual NED. For multilingual NED evaluation we are currently working on a German dataset, following the TAC-KBP dataset creation guidelines. In addition to Wikipedia, we also intend to exploit more dynamical information sources. For example, when considering news articles, NEs often occur for a certain period of time in consecutive news dealing with the same topic. This short-time context could be a useful source of information for disambiguating novel entities.

References

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Amsterdam, The Netherlands. Elsevier Science Publishers B. V.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 9–16, Trento, Italy.

Razvan Constantin Bunescu. 2007. *Learning for Information Extraction: From Named Entity Recognition and Disambiguation To Relation Extraction*. Ph.D. thesis, University of Texas at Austin, Department of Computer Sciences.

Silviu Cucerzan. 2007. Large-Scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic. Association for Computational Linguistics.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285, Beijing, China. Coling 2010 Organizing Committee.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

Xianpei Han and Jun Zhao. 2009. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 215–224, Hong Kong, China. ACM.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*.

Paul McNamee, Hoa Trang Dang, Heather Simpson, Patrick Schone, and Stephanie M. Strassel. 2010. An evaluation of technologies for knowledge base population. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

Vladimir N. Vapnik. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang. 2010. Entity linking leveraging automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1290–1298, Beijing, China. Coling 2010 Organizing Committee.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 483–491, Stroudsburg, PA, USA. Association for Computational Linguistics.

Extracting and Classifying Urdu Multiword Expressions

Annette Hautli
Department of Linguistics
University of Konstanz, Germany
annette.hautli@uni-konstanz.de

Sebastian Sulger
Department of Linguistics
University of Konstanz, Germany
sebastian.sulger@uni-konstanz.de

Abstract

This paper describes a method for automatically extracting and classifying multiword expressions (MWEs) for Urdu on the basis of a relatively small unannotated corpus (around 8.12 million tokens). The MWEs are extracted by an unsupervised method and classified into two distinct classes, namely locations and person names. The classification is based on simple heuristics that take the co-occurrence of MWEs with distinct postpositions into account. The resulting classes are evaluated against a hand-annotated gold standard and achieve an f-score of 0.5 and 0.746 for locations and persons, respectively. A target application is the Urdu ParGram grammar, where MWEs are needed to generate a more precise syntactic and semantic analysis.

1 Introduction

Multiword expressions (MWEs) are expressions which can be semantically and syntactically idiosyncratic in nature; acting as a single unit, their meaning is not always predictable from their components. Their identification is therefore an important task for any Natural Language Processing (NLP) application that goes beyond the analysis of pure surface structure, in particular for languages with few other NLP tools available.

There is a vast amount of literature on extracting and classifying MWEs automatically; many approaches rely on already available resources that aid during the acquisition process. In the case of the Indo-Aryan language Urdu, a lack of linguistic resources such as annotated corpora or lexical knowledge bases impedes the task of detecting and classifying MWEs. Nevertheless, statistical measures and language-specific syntactic information can be employed to extract and classify MWEs.

Therefore, the method described in this paper can partly overcome the bottleneck of resource sparsity, despite the relatively small size of the available corpus and the simplistic approach taken. With the help of heuristics as to the occurrence of Urdu MWEs with characteristic postpositions and other cues, it is possible to cluster the MWEs into two groups: locations and person names. It is also possible to detect junk MWEs. The classification is then evaluated against a hand-annotated gold standard of Urdu MWEs.

An NLP tool where the MWEs can be employed is the Urdu ParGram grammar (Butt and King, 2007; Bögel et al., 2007; Bögel et al., 2009), which is based on the Lexical-Functional Grammar (LFG) formalism (Dalrymple, 2001). For this task, different types of MWEs need to be distinguished as they are treated differently in the syntactic analysis.

The paper is structured as follows: Section 2 provides a brief review of related work, in particular on MWE extraction in Indo-Aryan languages. Section 3 describes our methodology, with the evaluation following in Section 4. Section 5 presents the Urdu ParGram Grammar and its treatment of MWEs, followed by the discussion and the summary of the paper in Section 6.

2 Related Work

MWE extraction and classification has been the focus of a large amount of research. However, much work

Proceedings of the ACL-HLT 2011 Student Session, pages 24–29,
Portland, OR, USA 19-24 June 2011.

has been conducted for well-resourced languages such as English, benefiting from large enough corpora (Attia et al., 2010), parallel data (Zarrieß and Kuhn, 2009) and NLP tools such as taggers or dependency parsers (Martens and Vandeghinste (2010), among others) and lexical resources (Pearce, 2001).

Related work on Indo-Aryan languages has mostly focused on the extraction of complex predicates, with the focus on Hindi (Mukerjee et al., 2006; Chakrabarti et al., 2008; Sinha, 2009) and Bengali (Das et al., 2010; Chakraborty and Bandyopadhyay, 2010). While complex predicates also make up a large part of the verbal inventory in Urdu (Butt, 1993), for the scope of this paper, we restrict ourselves to classifying MWEs as locations or person names and filter out junk bigrams.

Our approach deviates in several aspects to the related work in Indo-Aryan: First, we do not concentrate on specific POS constructions or dependency relations, but use an unannotated middle-sized corpus. For classification, we use simple heuristics by taking the postpositions of the MWEs into account. These can provide hints as to the nature of the MWE.

3 Methodology

3.1 Extraction and Identification of MWE Candidates

The bigram extraction was carried out on a corpus of around 8.12 million tokens of Urdu newspaper text, collected by the Center for Research in Urdu Language Processing (CRULP) (Hussain, 2008). We did not perform any pre-processing such as POS tagging or stop word removal.

Due to the relatively small size of our corpus, the frequency cut-off for bigrams was set to 5, i.e. all bigrams that occurred five times or more in the corpus were considered. This rendered a list of 172,847 bigrams which were then ranked with the X^2 association measure, using the UCS toolkit.[1]

The reasons for employing the X^2 association measure are twofold. First, papers using comparatively sized corpora reported encouraging results for similar experiments (Ramisch et al., 2008; Kizito et al., 2009). Second, initial manual comparison between MWE lists ranked according to all measures implemented in the UCS toolkit revealed the most convincing results for the X^2 test.

For the time being, we focus on bigram MWE extraction. While the UCS toolkit readily supports work on Unicode-based languages such as Urdu, it does not support trigram extraction; other freely available tools such as TEXT-NSP[2] do come with trigram support, but cannot handle Unicode script. As a consequence, we currently implement our own scripts to overcome these limitations.

3.2 Syntactic Cues

The clustering approach taken in this paper is based on Urdu-specific syntactic information that can be gathered straightforwardly from the corpus. Urdu has a number of postpositions that can be used to identify the nature of an MWE. Typographical cues such as initial capital letters do not exist in the Urdu script.

Locative postpositions The postposition پر (*par)* either expresses location on something which has a surface or that an object is next to something.[3] In addition, it expresses movement to a destination.

(1) نادیه تل ابیب پر گاي
nAdiyah t3ul AbEb par gAyI
Nadya Tel Aviv to go.Perf.Fem.Sg
'Nadya went to Tel Aviv.'

میں (*mEN*) expresses location in or at a point in space or time, whereas تك (*tak*) denotes that something extends to a specific point in space. سي (*sE*) shows movement away from a certain point in space.

These postpositions mostly occur with locations and are thus syntactic indicators for this type of MWE. However, in special cases, they can also occur with other nouns, in which case we predict wrong results during classification.

Person-indicating syntactic cues To classify an MWE as a person, we consider syntactic cues that usually occur after such MWEs. The ergative marker نے (*nE*) describes an agentive subject in transitive

[1] Available at `http://www.collocations.de`. See Evert (2004) for documentation.

[2] Available at `http://search.cpan.org/dist/Text-NSP`. See Banerjee and Pedersen (2003) for documentation.

[3] The employed transliteration scheme is explained in Malik et al. (2010).

	Locative			Instr.	Ergative	Possessive			Acc./Dat.
	پر (*par*)	میں (*mEN*)	تك (*tak*)	سي (*sE*)	ني (*nE*)	كا (*kA*)	كي (*kE*)	كي (*kI*)	كو (*kO*)
LOC	√	√	√	√	—	—	—	—	—
PERS	—	—	—	√	√	√	√	√	√
JUNK	—	—	—	—	—	—	—	—	—

Table 1: Heuristics for clustering Urdu MWEs by different postpositions

sentences; therefore, it forms part of our heuristic for finding person MWEs.

(2) نادیه نی یاسین کو مارا

nAdiyah nE yAsIn kO mArA
Nadya Erg Yasin Acc hit.Perf.Masc.Sg
'Nadya hit Yasin.'

The same holds for the possessive markers كا (*kA*), كي (*kE*) and كي (*kI*).

The accusative and dative case marker كو (*kO*) is also a possible indicator that the preceding MWE is a person.

These cues can also appear with common nouns, but the combination of MWE and syntactic cue hints to a person MWE. However, consider cases such as *New Delhi said that the taxes will rise.*, where *New Delhi* is treated as an agent with *nE* attached to it, providing a wrong clue as to the nature of the MWE.

3.3 Classifying Urdu MWEs

The classification of the extracted bigrams is solely based on syntactic information as described in the previous section. For every bigram, the postpositions that it occurs with are extracted from the corpus, together with the frequency of the co-occurrence.

Table 1 shows which postpositions are expected to occur with which type of MWE. The first stipulation is that only bigrams that occur with one of the locative postpositions plus the ablative/instrumental marker سي (*sE*) one or more times are considered to be locative MWEs (LOC). In contrast, bigrams are judged as persons (PERS) when they co-occur with all postpositions apart from the locative postpositions one or more times. If a bigram occurs with none of the postpositions, it is judged as being junk (JUNK). As a consequence this means that theoretically valid MWEs such as complex predicates, which never occur with a postposition, are misclassified as being JUNK.

Without any further processing, the resulting clusters are then evaluated against a hand-annotated gold standard, as described in the following section.

4 Evaluation

4.1 Gold Standard

Our gold standard comprises the 1300 highest ranked Urdu multiword candidates extracted from the CRULP corpus, using the X^2 association measure. The bigrams are then hand-annotated by a native speaker of Urdu and clustered into the following classes: locations, person names, companies, miscellaneous MWEs and junk. For the scope of this paper, we restrict ourselves to classifying MWEs as either locations or person names,. This also lies in the nature of the corpus: companies can usually be detected by endings such as "Corp." or "Ltd.", as is the case in English. However, these markers are often left out and are not present in the corpus at hand. Therefore, they cannot be used for our clustering. The class of miscellaneous MWEs contains complex predicates that we do not attempt to deal with here.

In total, the gold standard comprises 30 companies, 95 locations, 411 person names, 512 miscellaneous MWEs (mostly complex predicates) and 252 junk bigrams. We have not analyzed the gold standard any further, and restricting it to $n < 1300$ might improve the evaluation results.

4.2 Results

The bigrams are classified according to the heuristics outlined in Section 3.3. Evaluating against the hand-annotated gold standard yields the results in Table 2.

While the results are encouraging for persons with an f-score of 0.746, there is still room for improvement for locative MWEs. Part of the problem for per-

	Precision	Recall	F-Score	#total	#found
LOC	0.453	0.558	**0.5**	95	43
PERS	0.727	0.765	**0.746**	411	298
JUNK	0.472	0.317	**0.379**	252	119

Table 2: Results for MWE clustering

son names is that Urdu names are generally longer than two words, and as we have not considered trigrams yet, it is impossible to find a postposition after an incomplete though generally valid name. Locations tend to have the same problem, however the reasons for missing out on a large part of the locative MWEs are not quite clear and are currently being investigated.

Junk bigrams can be detected with an f-score of 0.379. Due to the heterogeneous nature of the miscellaneous MWEs (e.g., complex predicates), many of them are judged as being junk because they never occur with a postposition. If one could detect complex predicate and, possibly, other subgroups from the miscellaneous class, then classifying the junk MWEs would become easier.

5 Integration into the Urdu ParGram Grammar

The extracted MWEs are integrated into the Urdu ParGram grammar (Butt and King, 2007; Bögel et al., 2007; Bögel et al., 2009), a computational grammar for Urdu running with XLE (Crouch et al., 2010) and based on the syntax formalism of LFG (Dalrymple, 2001). XLE grammars are generally handwritten and not acquired a machine learning process or the like. This makes grammar development a very conscious task and it is imperative to deal with MWEs in order to achieve a linguistically valid and deep syntactic analysis that can be used for an additional semantic analysis.

MWEs that are correctly classified according to the gold standard are automatically integrated into the multiword lexicon of the grammar, accompanied by information about their nature (see example (3)).

In general, grammar input is first tokenized by a standard tokenizer that separates the input string into single tokens and replaces the white spaces with a special token boundary symbol. Each token is then passed through a cascade of finite-state morphological analyzers (Beesley and Karttunen, 2003). For MWEs, the matter is different as they are treated as a single unit to preserve the semantic information they carry. Apart from the meaning preservation, integrating MWEs into the grammar reduces parsing ambiguity and parsing time, while the perspicuity of the syntactic analyses is increased (Butt et al., 1999).

In order to prevent the MWEs from being independently analyzed by the finite-state morphology, a look-up is performed in a transducer which only contains MWEs with their morphological information. So instead of analyzing *t3ul* and *AbEb* separately, for example, they are analyzed as a single item carrying the morphological information `+Noun+Location`.[4]

(3) `t3ul` AbEb: /`t3ul` AbEb/ +Noun +Location

The resulting stem and tag sequence is then passed on to the grammar. See (4) for an example and Figures 1 and 2 for the corresponding c- and f-structure; the `+Location` tag in (3) is used to produce the location analysis in the f-structure. Note also that `t3ul AbEb` is displayed as a multiword under the `N` node in the c-structure.

(4) نادیه تل ابیب پر گاي
nAdiyah t3ul AbEb par gAyI
Nadya Tel Aviv to go.Perf.Fem.Sg
'Nadya went to Tel Aviv.'

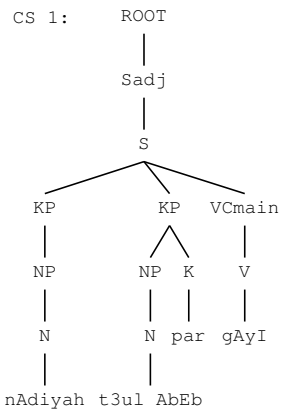

Figure 1: C-structure for (4)

[4]The ` symbol is an escape character, yielding a literal white space.

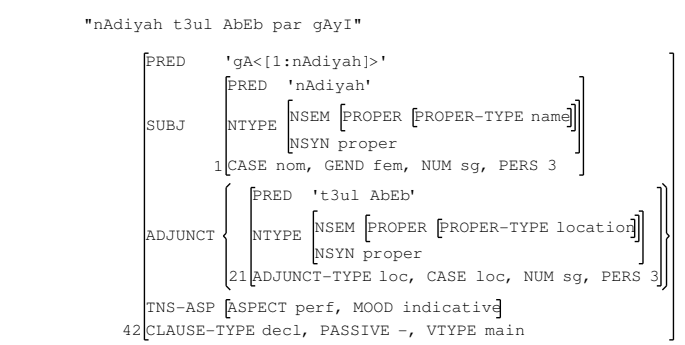

Figure 2: F-structure for (4)

6 Discussion, Summary and Future Work

Despite the simplistic approach for extracting and clustering Urdu MWEs taken in this paper, the results are encouraging with f-scores of 0.5 and 0.746 for locations and person names, respectively. We are well aware that this paper does not present a complete approach to classifying Urdu multiwords, but considering the targeted tool, the Urdu ParGram grammar, this methodology provides us with a set of MWEs that can be implemented to improve the syntactic analyses.

The methodology provided here can also guide MWE work in other languages facing the same resource sparsity as Urdu, given that distinctive syntactic cues are available in the language.

For Urdu, the syntactic cues are good indications of the nature of the MWE; future work on this subtopic might prove beneficial to the clustering regarding companies, complex predicates and junk MWEs. Another area for future work is to extend the extraction and classification to trigrams to improve the results especially for locations and person names. We also consider harvesting data sources from the web such as lists of cities, common names and companies in Pakistan and India. Such lists are not numerous for Urdu, but they may nevertheless help to generate a larger MWE lexicon.

Acknowledgments

We would like to thank Samreen Khan for annotating the gold standard, as well as the anonymous reviewers for their valuable comments. This research was in part supported by the Deutsche Forschungsgemeinschaft (DFG).

References

Mohammed Attia, Antonio Toral, Lamia Tounsi, Pavel Pecina, and Josef van Genabith. 2010. Automatic Extraction of Arabic Multiword Expressions. In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*.

Satanjeev Banerjee and Ted Pedersen. 2003. The Design, Implementation and Use of the Ngram Statistics Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*.

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

Tina Bögel, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2007. Developing a Finite-State Morphological Analyzer for Urdu and Hindi: Some Issues. In *Proceedings of FSMNLP07, Potsdam, Germany*.

Tina Bögel, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2009. Urdu and the Modular Architecture of ParGram. In *Proceedings of the Conference on Language and Technology 2009 (CLT09)*.

Miriam Butt and Tracy Holloway King. 2007. Urdu in a Parallel Grammar Development Environment. *Language Resources and Evaluation*, 41(2):191–207.

Miriam Butt, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. CSLI Publications.

Miriam Butt. 1993. *The Structure of Complex Predicates in Urdu*. Ph.D. thesis, Stanford University.

Debasri Chakrabarti, Vaijayanthi M. Sarma, and Pushpak Bhattacharyya. 2008. Hindi Compound Verbs and their Automatic Extraction. In *Proceedings of COLING 2008*, pages 27–30.

Tanmoy Chakraborty and Sivaji Bandyopadhyay. 2010. Identification of Reduplication in Bengali Corpus and their Semantic Analysis: A Rule-Based Approach. In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, pages 72–75.

Dick Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman, 2010. *XLE Documentation*. Palo Alto Research Center.

Mary Dalrymple. 2001. *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*. Academic Press.

Dipankar Das, Santanu Pal, Tapabrata Mondal, Tanmoy Chakraborty, and Sivaji Bandyopadhyay. 2010. Automatic Extraction of Complex Predicates in Bengali. In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, pages 37–45.

Stefan Evert. 2004. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, IMS, University of Stuttgart.

Sarmad Hussain. 2008. Resources for Urdu Language Processing. In *Proceedings of the 6th Workshop on Asian Language Resources, IJCNLP'08*.

John Kizito, Ismail Fahmi, Erik Tjong Kim Sang, Gosse Bouma, and John Nerbonne. 2009. Computational Linguistics and the History of Science. In Liborio Dibattista, editor, *Storia della Scienza e Linguistica Computazionale*. FrancoAngeli.

Muhammad Kamran Malik, Tafseer Ahmed, Sebastian Sulger, Tina Bögel, Atif Gulzar, Ghulam Raza, Sarmad Hussain, and Miriam Butt. 2010. Transliterating Urdu for a Broad-Coverage Urdu/Hindi LFG Grammar. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*.

Scott Martens and Vincent Vandeghinste. 2010. An Efficient, Generic Approach to Extracting Multi-Word Expressions from Dependency Trees. In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, pages 84–87.

Amitabha Mukerjee, Ankit Soni, and Achla M. Raina. 2006. Detecting Complex Predicates in Hindi using POS Projection across Parallel Corpora. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties (MWE '06)*, pages 28–35.

David Pearce. 2001. Synonymy in Collocation Extraction. In *WordNet and Other Lexical Resources: Applications, Extensions & Customizations*, pages 41–46.

Carlos Ramisch, Paulo Schreiner, Marco Idiart, and Aline Villavicencio. 2008. An Evaluation of Methods for the Extraction of Multiword Expressions. In *Proceedings of the Workshop on Multiword Expressions: Towards a Shared Task for Multiword Expressions (MWE 2008)*.

R. Mahesh K. Sinha. 2009. Mining Complex Predicates in Hindi Using a Parallel Hindi-English Corpus. In *Proceedings of the 2009 Workshop on Multiword Expressions, ACL-IJCNLP 2009*, pages 40–46.

Sina Zarrieß and Jonas Kuhn. 2009. Exploiting Translational Correspondences for Pattern-Independent MWE Identification. In *Proceedings of the 2009 Workshop on Multiword Expressions, ACL-IJCNLP 2009*, pages 23–30.

A Latent Topic Extracting Method based on Events in a Document and its Application

Risa Kitajima
Ochanomizu University
kitajima.risa@is.ocha.ac.jp

Ichiro Kobayashi
Ochanomizu University
koba@is.ocha.ac.jp

Abstract

Recently, several latent topic analysis methods such as LSI, pLSI, and LDA have been widely used for text analysis. However, those methods basically assign topics to words, but do not account for the events in a document. With this background, in this paper, we propose a latent topic extracting method which assigns topics to events. We also show that our proposed method is useful to generate a document summary based on a latent topic.

1 Introduction

Recently, several latent topic analysis methods such as Latent Semantic Indexing (LSI) (Deerwester et al., 1990), Probabilistic LSI (pLSI) (Hofmann, 1999), and Latent Dirichlet Allocation (LDA) (Blei et al., 2003) have been widely used for text analysis. However, those methods basically assign topics to words, but do not account for the events in a document. Here, we define a unit of informing the content of document at the level of sentence as an "Event" [1], and propose a model that treats a document as a set of Events. We use LDA as a latent topic analysis method, and assign topics to Events in a document. To examine our proposed method's performance on extracting latent topics from a document, we compare the accuracy of our method to that of the conventional methods through a common document retrieval task. Furthermore, as an application of our method, we apply it to a query-biased document summarization (Tombros and Sanderson, 1998; Okumura and Mochizuki, 2000; Berger and Mittal, 2000) to verify that the method is useful for various applications.

2 Related Studies

Suzuki et al. (2010) proposed a flexible latent topics inference in which topics are assigned to phrases in a document. Matsumoto et al. (2005) showed that the accuracy of document classification will be improved by introducing a feature dealing with the dependency relationships among words.

In case of assigning topics to words, it is likely that two documents, which have the same word frequency in themselves, tend to be estimated as they have the same topic probablistic distribution without considering the dependency relation among words. However, there are many cases where the relationship among words is regarded as more important rather than the frequency of words as the feature identifying the topics of a document. For example, in case of classifying opinions to objects in a document, we have to identify what sort of opinion is assigned to the target objects, therefore, we have to focus on the relationship among words in a sentence, not only on the frequent words appeared in a document. For this reason, we propose a method to assign topics to Events instead of words.

As for studies on document summarization, there are various methods, such as the method based on word frequency (Luhn, 1958; Nenkova and Vanderwende, 2005), and the method based on a graph (Radev, 2004; Wan and Yang, 2006). Moreover, several methods using a latent topic model have been proposed (Bing et al., 2005; Arora and Ravin-

[1] For the definition of an Event, see Section 3.

Proceedings of the ACL-HLT 2011 Student Session, pages 30–35,
Portland, OR, USA 19-24 June 2011.

dran, 2008; Bhandari et al., 2008; Henning, 2009; Haghighi and Vanderwende, 2009). In those studies, the methods estimate a topic distribution on each sentence in the same way as the latent semantic analysis methods normally do that on each document, and generate a summary based on the distribution. We also show that our proposed method is useful for the document summarization based on extracting latent topics from sentences.

3 Topic Extraction based on Events

In this study, since we deal with a document as a set of Events, we extract Events from each document; define some of the extracted Events as the index terms for the whole objective documents; and then make an Event-by-document matrix consisting of the frequency of Events to the documents. A latent topic distribution is estimated based on this matrix.

3.1 Definition of an Event

In this study, we define a pair of words in dependent relation which meets the following conditions: (Subject, Predicate) or (Predicate1, Predicate2) , as an Event. A noun and unknown words correspond to Subject, while a verb, adjective and adjective verb correspond to Predicate. To extract these pairs, we analyze the dependency structure of sentences in a document by a Japanese dependency structure analyzer, CaboCha [2]. The reason why we define (Predicete1, Predicate2) as an Event is because we recognized the necessity of such type of an Event by investigating the extracted pairs of words and comparing them with the content of the target document in preliminary experiments, and could not extract any Event in case of extracting an Event from the sentences without subject.

3.2 Making an Event-by-Document Matrix

In making a word-by-document matrix, high-frequent words appeared in any documents, and extremely infrequent words are usually not included in the matrix. In our method, high-frequent Events like the former case were not observed in preliminary experiments. We think the reason for this is because an Event, a pair of words, can be more meaningful than a single word, therefore, an Event is particularly a good feature to express the meaning of a document. Meanwhile, the average number of Events per sentence is 4.90, while the average number of words per sentence is 8.93. A lot of infrequent Events were observed in the experiments because of the nature of an Event, i.e., a pair of words. This means that the same process of making a word-by-document matrix cannot be applied to making an Event-by-document matrix because the nature of an Event as a feature expressing a document is different from that of a word. In concrete, if the events, which once appear in documents, would be removed from the candidates to be a part of a document vector, there might be a case where the constructed document vector does not reflect the content of the original documents. Considering this, in order to make the constructed document vector reflect the content of the original documents, we do not remove the Event only itself extracted from a sentence, even though it appears only once in a document.

3.3 Estimating a Topic Distribution

After making an Event-by-document matrix, a latent topic distribution of each Event is estimated by means of Latent Dirichlet Allocation. Latent Dirichlet Allocation is a generative probabilistic model that allows multiple topics to occur in a document, and gets the topic distribution based on the idea that each topic emerges in a document based on a certain probability. Each topic is expressed as a multinomial distribution of words.

In this study, since a topic is assigned to an Event, each topic is expressed as a multinomial distribution of Events. As a method to estimate a topic distribution, while a variational Bayes method (Blei et al., 2003) and its application (Teh et al., 2006) have been proposed, in this study we use Gibbs sampling method (Grififths and Steyvers, 2004). Furthermore, we define a sum of topic distributions of the events in a query as the topic distribution of the query.

4 Performance Evaluation Experiment

Through a common document retrieval task, we compare our method with the conventional method and evaluate both of them. In concrete, we regard the documents which have a similar topic distribu-

[2] http://chasen.org/ taku/software/cabocha/

tion to a query's topic distribution as the result of retrieval, and then examine whether or not the estimated topic distribution can represent the latent semantics of each document based on the accuracy of retrieval results. Henceforth, we call the conventional word-based LDA "wordLDA" and our proposed event-based LDA "eventLDA".

4.1 Measures for Topic Distribution

As measures for identifying the similarity of topic distribution, we adopt Kullback-Leibler Divergence (Kullback and Leibler, 1951), Symmetric Kullback-Leibler Divergence (Kullback and Leibler, 1951), Jensen-Shannon Divergence (Lin, 2002), and cosine similarity. As for wordLDA, Henning (2009) has reported that Jensen-Shannon Divergence shows the best performance among the above measures in terms of estimating the similarity between two sentences. We also compare the performance of the above measures when using eventLDA.

4.2 Experimental Settings

As for the documents used in the experiment, we use a set of data including users' reviews and their evaluations for hotels and their facilities, provided by Rakuten Travel[3]. Each review has five-grade evaluations of a hotel's facilities such as room, location, and so on. Since the data hold the relationships between objects and their evaluations, therefore, it is said that they are appropriate for the performance evaluation of our method because the relationship is usually expressed in a pair of words, i.e., an Event. The query we used in the experiment was "a room is good". The total number of documents is 2000, consisting of 1000 documents randomly selected from the users' reviews whose evaluation for "a room" is 1 (bad) and 1000 documents randomly selected from the reviews whose evaluation is 5 (good). The latter 1000 documents are regarded as the objective documents in retrieval. Because of this experiment design, it is clear that the random choice for retrieving "good" vs. "bad" is 50%. As for the evaluation measure, we adopt 11-point interpolated average precision.

In this experiment, a comparison between the both methods, i.e., wordLDA and eventLDA, is conducted from the viewpoints of the proper number of topics and the most useful measure to estimate similarity. At first, we use Jensen-Shannon Divergence as the measure to estimate the similarity of topic distribution, changing the number of topics k in the following, $k = 5$, $k = 10$, $k = 20$, $k = 50$, $k = 100$, and $k = 200$. Next, the number of topics is fixed based on the result of the first process, and then it is decided which measure is the most useful by applying each measure to estimate the similarity of topic distributions. Here, the iteration count of Gibbs Sampling is 200. The number of trials is 20, and all trials are averaged. The same experiment is conducted for wordLDA to compare both results.

4.3 Result

Table 1 shows the retrieval result examined by 11-point interpolated average precision, changing the number of topics k. High accuracy is shown at $k = 5$ in eventLDA, and $k = 50$ in wordLDA, respectively. Overall, we see that eventLDA keeps higher accuracy than wordLDA.

number of topics	wordLDA	eventLDA
5	0.5152	0.6256
10	0.5473	0.5744
20	0.5649	0.5874
50	0.5767	0.5740
100	0.5474	0.5783
200	0.5392	0.5870

Table 1: Result based on the number of topics.

Table 2 shows the retrieval result examined by 11-point interpolated average precision under various measures. The number of topics k is $k = 50$ in wordLDA and $k = 5$ in eventLDA respectively, based on the above result. Under any measures, we see that eventLDA keeps higher accuracy than wordLDA.

similarity measure	wordLDA	eventLDA
Kullback-Leibler	0.5009	0.5056
Symmetric Kullback-Leibler	0.5695	0.6762
Jensen-Shannon	0.5753	0.6754
cosine	0.5684	0.6859

Table 2: Performance under various measures.

4.4 Discussions

The result of the experiment shows that eventLDA provides a better performance than wordLDA, there-

[3]http://travel.rakuten.co.jp/

fore, we see our method can properly treat the latent topics of a document. In addition, as for a property of eventLDA, we see that it can provide detail classification with a small number of topics. As the reason for this, we think that a topic distribution on a feature is narrowed down to some extent by using an Event as the feature instead of a word, and then as a result, the possibility of generating error topics decreased.

On the other hand, a proper measure for our method is identified as cosine similarity, although cosine similarity is not a measure to estimate probabilistic distribution. It is unexpected that the measures proper to estimate probabilistic distribution got the result of lower performance than cosine similarity. From this, there are some space where we need to examine the characteristics of topic distribution as a probabilistic distribution.

5 Application to Summarization

Here, we show multi-document summarization as an application of our proposed method. We make a query-biased summary, and show the effectiveness of our method by comparing the accuracy of a generated summary by our method with that of summaries by the representative summarization methods often used as benchmark methods to compare.

5.1 Extracting Sentences by MMR-MD

In extracting important sentences, considering only similarity to a given query, we may generate a redundant summary. To avoid this problem, a measure, MMR-MD (Maximal Marginal Relevance Multi-Document), was proposed (Goldstein et al., 2000). This measure is the one which prevents extracting similar sentences by providing penalty score that corresponds to similarity between a newly extracted sentence and the previously extracted sentences. It is defined by Eq. 1 (Okumura and Nanba, 2005).

$$MMR\text{-}MD \equiv argmax_{C_i \in R \setminus S}[\lambda Sim_1(C_i,Q) - (1-\lambda) max_{C_j \in S} Sim_2(C_i,C_j)] \quad (1)$$

We aim to choose sentences whose content is similar to query's content based on a latent topic, while reducing the redundancy of choosing similar sentences to the previously chosen sentences. Therefore, we adopt the similarity of topic distributions for Sim_1 which estimates similarity between a sentence and a query, and adopt cosine similarity based on Events as a feature unit for Sim_2 which estimates the similarity with the sentences previously chosen. As the measures to estimate topic distribution similarity, we use the four measures explained in Section 4.1. Here, as for the weighting parameter λ, we set $\lambda = 0.5$.

C_i : sentence in the document sets
Q : query
R : a set of sentences retrieved by Q from the document sets
S : a set of sentences in R already extracted
λ : weighting parameter

5.2 Experimental Settings

In the experiment, we use a data set provided at NTCIR4 (NII Test Collection for IR Systems 4) TSC3 (Text Summarization Challenge 3) [4] .

The data consists of 30 topic sets of documents in which each set has about 10 Japanese newspaper articles, and the total number of the sentences in the data is 3587. In order to make evaluation for the result provided by our method easier, we compile a set of questions, provided by the data sets for evaluating the result of summarization, as a query, and then use it as a query for query-biased summarization. As an evaluation method, we adopt precision and coverage used at TSC3 (Hirao et al., 2004), and the number of extracted sentences is the same as used in TSC3. Precision is an evaluation measure which indicates the ratio of the number of correct sentences to that of the sentences generated by the system. Coverage is an evaluation measure which indicates the degree of how the system output is close to the summary generated by a human, taking account of the redundancy.

Moreover, to examine the characteristics of the proposed method, we compare both methods in terms of the number of topics and the proper measure to estimate similarity. The number of trials is 20 at each condition. 5 sets of documents selected at random from 30 sets of documents are used in the trials, and all the trials are totally averaged. As a target for comparison with the proposed method, we also conduct an experiment using wordLDA.

[4] http://research.nii.ac.jp/ntcir/index-en.html

5.3 Result

As a result, there is no difference among the four measures — the same result is obtained by the four measures. Table 3 shows comparison between eventLDA and wordLDA in terms of precision and coverage. The number of topics providing the highest accuracy is $k = 5$ for wordLDA, and $k = 10$ for eventLDA, respectively.

number of topics	wordLDA		eventLDA	
	Precision	Coverage	Precision	Coverage
5	0.314	0.249	0.404	0.323
10	0.264	0.211	0.418	0.340
20	0.261	0.183	0.413	0.325
50	0.253	0.171	0.392	0.319

Table 3: Comparison of the number of topics.

Furthermore, Table 4 shows comparison between the proposed method and representative summarization methods which do not deal with latent topics. As representative summarization methods to compare our method, we took up the Lead method (Brandow et al., 1995) which is effective for document sumarization of newspapers, and the important sentence extraction-based summarization method using TF-IDF.

method	Precision	Coverage
Lead	0.426	0.212
TF-IDF	0.454	0.305
wordLDA (k=5)	0.314	0.249
eventLDA (k=10)	0.418	0.340

Table 4: Comparison of each method.

5.4 Discussions

Under any condition, eventLDA provides a higher accuracy than wordLDA. We see that the proposed method is useful for estimating a topic on a sentence. As the reason for that the accuracy does not depend on any kinds of similarity measures, we think that an estimated topic distribution is biased to a particular topic, therefore, there was not any influence due to the kinds of similarity measures. Moreover, the proper number of topics of eventLDA is bigger than that of wordLDA. We consider the reason for this is because we used newspaper articles as the objective documents, so it can be thought that the topics onto the words in the articles were specific to some extent; in other words, the words often used in a particular field are often used in newspaper articles, therefore, we think that wordLDA can classify the documents with the small number of topics. In comparison with the representative methods, the proposed method takes close accuracy to their accuracy, therefore, we see that the performance of our method is at the same level as those representative methods which directly deal with words in documents. In particular, as for coverage, our method shows high accuracy. We think the reason for this is because a comprehensive summary was made by latent topics.

6 Conclusion

In this paper, we have defined a pair of words with dependency relationship as "Event" and proposed a latent topic extracting method in which the content of a document is comprehended by assigning latent topics onto Events. We have examined the ability of our proposed method in Section 4, and as its application, we have shown a document summarization using the proposed method in Section 5. We have shown that eventLDA has higher ability than wordLDA in terms of estimating a topic distribution on even a sentence or a document; furthermore, even in case of assigning a topic on an Event, we see that latent topics can be properly estimated. Since an Event can hold a relationship between a pair of words, it can be said that our proposed method, i.e., eventLDA, can comprehend the content of a document more deeper and proper than the conventional method, i.e., wordLDA. Therefore, eventLDA can be effectively applied to various document data sets rather than wordLDA can be. We have also shown that another feature other than a word, i.e., an Event is also useful to estimate latent topics in a document. As future works, we will conduct experiments with various types of data and query, and further investigate the characteristic of our proposed method.

Acknowledgments

We would like to thank Rakuten, Inc. for permission to use the resources of Rakuten Travel, and thank the National Institute of Informatics for providing NTCIR data sets.

References

Adam Berger and Vibhu O. Mittal. 2000. Query-relevant summarization using FAQs. In *ACL '00 Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*:294–301.

Anastasios Tombros and Mark Sanderson. 1998. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*:2–10.

Ani Nenkova and Lucy Vanderwende. 2005. The Impact of Frequency on Summarization. Technical report, Microsoft Research.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring Content Models for Multi-Document Summarization. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*:362–370.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*,3:993–1022.

Dragomir R. Radev. 2004. Lexrank: graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR.*

Harendra Bhandari, Masashi Shimbo, Takahiko Ito, and Yuji Matsumoto. 2008. Generic Text Summarization Using Probabilistic Latent Semantic Indexing. In *Proceedings of the 3rd International Joint Conference on Natural Langugage Proceeding*:133-140.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAALP-ANLP Workshop on Automatic Summarization*:40–48.

Jianhua Lin. 2002. Divergence Measures based on the Shannon Entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Leonhard Henning. 2009. Topic-based Multi-Document Summarization with Probabilistic Latent Semantic Analysis. *Recent Advances in Natural Language Processing*:144–149.

Manabu Okumura and Eiji Nanba. 2005. *Science of knowledge: Automatic Text Summarization.*(in Japanese) ohmsha.

Manabu Okumura and Hajime Mochizuki. 2000. Query-Biased Summarization Based on Lexical Chaining. *Computational Intelligence*,16(4):578–585.

Qin Bing, Liu Ting, Zhang Yu, and Li Sheng. 2005. Research on Multi-Document Summarization Based on Latent Semantic Indexing. *Journal of Harbin Institute of Technology*,12(1):91–94.

Rachit Arora and Balaraman Ravindran. 2008. Latent dirichlet allocation based multi-document summarization. In *Proceedings of the 2nd Workshop on Analytics for Noisy Unstructured Text Data.*

Ronald Brandow, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management: an International Journal - Special issue: summarizing text*,31(5):675–685.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. 2005. Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees. In *Proceedings of the 9th Pacific-Asia International Conference on Knowledge Discovery and Data Mining*:301–310.

Solomon Kullback and Richard A. Leibler. 1951. On Information and Sufficiency. *Annuals of Mathematical Statistics*, 22:49–86.

Thomas L. Grififths and Mark Steyvers. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*,101:5228–5235.

Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*:50–57.

Tsutomu Hirao, Takahiro Fukusima, Manabu Okumura, Chikashi Nobata, and Hidetsugu Nanba. 2004. Corpus and evaluation measures for multiple document summarization with multiple sources. In *Proceedings of the 20th International Conference on Computational Linguistics*:535–541.

Xiaojun Wan and Jianwu Yang. 2006. Improved affinity graph based multi-document summarization. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*

Yasuhiro Suzuki, Takashi Uemura, Takuya Kida, and Hiroki Arimura. 2010. Extension to word phrase on latent dirichlet allocation. Forum on Data Engineering and Information Management,i-6.

Yee W. Teh, David Newman, and Max Welling. 2006. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems Conference*,19:1353–1360.

Syntax-based Statistical Machine Translation using Tree Automata and Tree Transducers

Daniel Emilio Beck
Computer Science Department
Federal University of São Carlos
daniel_beck@dc.ufscar.br

Abstract

In this paper I present a Master's thesis proposal in syntax-based Statistical Machine Translation. I propose to build discriminative SMT models using both *tree-to-string* and *tree-to-tree* approaches. Translation and language models will be represented mainly through the use of Tree Automata and Tree Transducers. These formalisms have important representational properties that makes them well-suited for syntax modeling. I also present an experiment plan to evaluate these models through the use of a parallel corpus written in English and Brazilian Portuguese.

1 Introduction

Statistical Machine Translation (SMT) has dominated Machine Translation (MT) research in the last two decades. One of its variants, Phrase-based SMT (PB-SMT), is currently considered the state of the art in the area. However, since the advent of PB-SMT by Koehn et al. (2003) and Och and Ney (2004), purely statistical MT systems have not achieved considerable improvements. So, new research directions point toward the use of linguistic resources integrated into SMT systems.

According to Lopez (2008), there are four steps when building an SMT system: *translational equivalence modeling*[1]*, parameterization, parameter estimation* and *decoding*. This Master's thesis proposal aims to improve SMT systems by including syntactic information in the first and second steps. Therefore, I plan to investigate two approaches: the Tree-to-String (TTS) and the Tree-to-Tree (TTT) models. In the former, syntactic information is provided only for the source language while in the latter, it is provided for both source and target languages.

There are many formal theories to represent syntax in a language, like Context-free Grammars (CFGs), Tree Substitution Grammars (TSGs), Tree Adjoining Grammars (TAGs) and all its synchronous counterparts. In this work, I represent each sentence as a constituent tree and use Tree Automata (TAs) and Tree Transducers (TTs) in the language and translation models.

Although this work is mainly language independent, proof-of-concept experiments will be executed on the English and Brazilian Portuguese (en-ptBR) language pair. Previous research on factored translation for this pair (using morphological information) showed that it improved the results in terms of BLEU (Papineni et al., 2001) and NIST (Doddington, 2002) scores, as shown in Table 1 (Caseli and Nunes, 2009). However, even factored translation models have limitations: many languages (and Brazilian Portuguese is not an exception) have relatively loose word order constraints and present long-distance agreements that cannot be efficiently represented by those models. Such phenomena motivate the use of more powerful models that take syntactic information into account.

2 Related work

Syntax-based approaches for SMT have been proposed in many ways. Some apply the TTS model: Yamada and Knight (2001) uses explicit inser-

[1] For the remainder of this proposal, I will refer to this step as simply *translation model*.

Proceedings of the ACL-HLT 2011 Student Session, pages 36–40,
Portland, OR, USA 19-24 June 2011.

	en-ptBR		ptBR-en	
	BLEU	NIST	BLEU	NIST
PB-SMT	0,3589	7,8312	0,3903	8,3008
FT	0,3713	7,9813	0,3932	8,4421

Table 1: BLEU and NIST scores for PB-SMT and factored translation experiments for the en-ptBR language pair

tion, reordering and translation rules, Nguyen et al. (2008) uses synchronous CFGs rules and Liu et al. (2006) uses TTs. Galley et al. (2006) also uses transducer rules but extract them from parse trees in target language instead (the *string-to-tree* approach - STT). Works that apply the TTT model include Gildea (2003) and Zhang et al. (2008). All those works also include methods and algorithms for efficient rule extraction since it's unfeasible to extract all possible rules from a parsed corpus due to exponential cost.

There have been research efforts to combine syntax-based systems with phrase-based systems. These works mainly try to incorporate non-syntatic phrases into a syntax-based model: while Liu et al. (2006) integrates bilingual phrase tables as separate TTS templates, Zhang et al. (2008) uses an algorithm to convert leaves in a parse tree to phrases before rule extraction.

Language models that take into account syntactic aspects have also been an active research subject. While works like Post and Gildea (2009) and Vandeghinste (2009) focus solely on language modeling itself, Graham and van Genabith (2010) shows an experiment that incorporates a syntax-based model into an PB-SMT system.

3 Tree automata and tree transducers

Tree Automata are similar to Finite-state Automata (FSA), except they recognize trees instead of strings (or sequences of words). Formally, FSA can only represent Regular Languages and thus, cannot efficiently model several syntactic features, including long-distance agreement. TA recognize the so-called Regular Tree Languages (RTLs), which can represent Context-free Languages (CFLs) since a set of all syntactic trees of a CFL is an RTL (Comon et al., 2007). However, it is important to note that the reciprocal is not true: there are RTLs that cannot be modeled by a CFL because those cannot capture the inner structure of trees. Figure 1 shows such an RTL, composed of two trees. If we extract an CFG from this RTL it would have the recursive rule S $\rightarrow$ SS, which would generate an infinite set of syntactic trees. In other words, there isn't an CFG capable to generate only the syntactic trees contained in the RTL shown in Figure 1. This feature implies that RTLs have more representational power than CFLs.

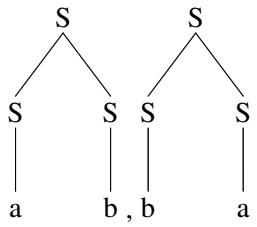

Figure 1: An RTL that cannot be modeled by a CFL

As a Finite-state Transducer (FST) is an extension of an FSA that produces strings, a Tree Transducer is an extension of a TA that produces trees. An FST is composed by an input RTL, an output RTL and a set of transformation rules. Restrictions can be added to the rules, leading to many TT variations, each with its properties (Graehl et al., 2008). The variations studied in this work are the *xT* (*extended top-down*, for TTT models) and *xTS* (*extended top-down tree-to-string*, for TTS models).

Top-down (T) transducers processes input trees starting from its root and descending through its nodes until it reaches the leaves, in contrast to *bottom-up* transducers, which do the opposite. Figure 2 shows a T rule, where uppercase letters (*NP*) represent symbols, lowercase letters (*q, r, s*) represent states and *x1* and *x2* are variables (formal definitions can be found in Comon et al. (2007)). Default top-down transducers must have only one symbol on the left-hand sides and thus cannot model some syntactic transformations (like local reordering, for example) without relying on copy and delete operations (Maletti et al., 2009). Extended top-down transducers allow multiple symbols on left-hand sides, making them more suited for syntax modeling. This property is shown on Figure 3 (adapted from Maletti et al. (2009)). Tree-to-string transducers simply drop the tree structure on right-

hand sides, which makes them adequate for translation models wihtout syntactic information in one of the languages. Figure 4 shows an example of a xTS rule, applied for the en-ptBR pair.

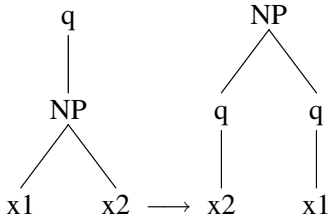

Figure 2: Example of a T rule

4 SMT Model

The systems will be implemented using a discriminative, log-linear model (Och and Ney, 2002), using the language and translation models as feature functions. Settings that uses more features besides those two models will also be built. In particular, I will investigate settings that incorporate non-syntactic phrases, using methods similar to Liu et al. (2006) and Zhang et al. (2008)

The translation models will be *weighted* TTs (Graehl et al., 2008), which add probabilities to the rules. These probabilities will be learned by an EM algorithm similar to the one described in Graehl et al. (2008). Rule extraction for TTS will be similar to the GHKM algorithm described in Galley et al. (2004) but I also plan to investigate the approaches used by Liu et al. (2006) and Nguyen et al. (2008). For TTT rule extraction, I will use a method similar to the one described in Zhang et al. (2008).

I also plan to use language models which takes into account syntactic properties. Although most works in syntactic language models uses tree grammars like TSGs and TAGs, these can be simulated by TAs and TTs (Shieber, 2004; Maletti, 2010). This property can help the systems implementation because it's possible to unite language and translation modeling in one TT toolkit.

5 Methods

In this section, I present the experiments proposed in my thesis and the materials required, along with the metrics used for evaluation. This work is planned to be done over a year.

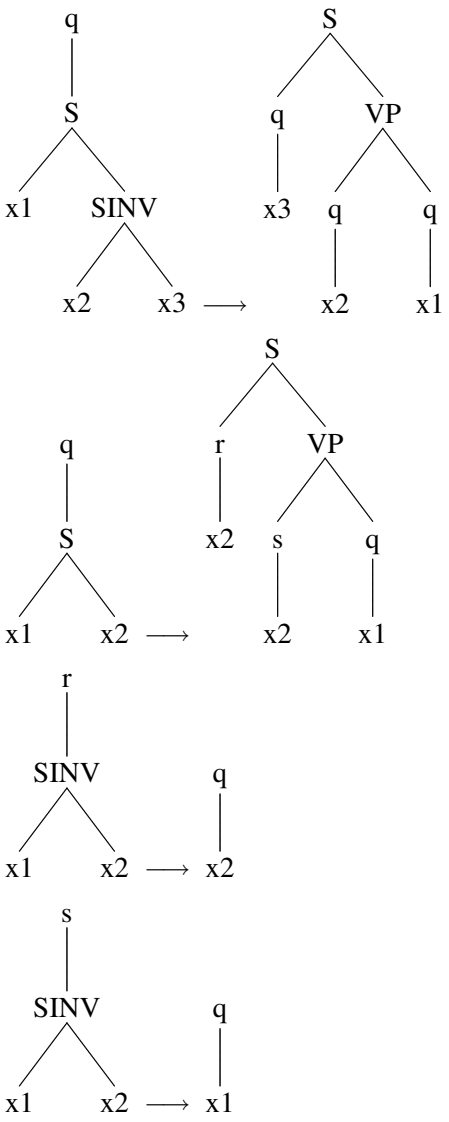

Figure 3: Example of a xT rule and its corresponding T rules

5.1 Materials

To implement and evaluate the techniques described, a parallel corpus with syntactic annotation is required. As the focus of this thesis is the English and Brazilian Portuguese language pair, I will use the PesquisaFAPESP corpus[2] in my experiments. This corpus is composed of 646 scientific papers, originally written in Brazilian Portuguese and manually translated into English, resulting in about 17,000 parallel sentences. As for syntactic annotation, I will use the Berkeley parser (Petrov and Klein, 2007) for

[2] http://revistapesquisa.fapesp.br

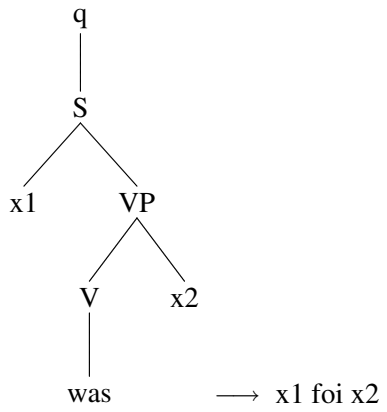

Figure 4: Example of a xTS rule (for the en-ptBR language pair)

English and the PALAVRAS parser (Bick, 2000) for Brazilian Portuguese.

In addition to the corpora and parsers, the following tools will be used:

- GIZA++[3] (Och and Ney, 2000) for lexical alignment
- Tiburon[4] (May and Knight, 2006) for transducer training in both TTS and TTT systems
- Moses[5] (Koehn et al., 2007) for decoding

5.2 Experiments and evaluation

Initially the corpus will be parsed using the tools described in section 5.1 and divided into a training set and a test set. For the TTS systems (one for each translation direction), the training set will be lexically aligned using GIZA++ and for the TTT system, its syntactic trees will be aligned using techniques similar to the ones proposed by Gildea (2003) and by Zhang et al. (2008). Both TTS and TTT systems will be implemented using Tiburon and Moses. For evaluation, BLEU and NIST scores on the test set will be used. The baseline will be the score for factored translation, shown in Table 1.

6 Contributions

After its conclusion, this thesis will have brought the following contributions:

- Language-independent SMT models which incorporates syntactic information in both language and translation models.
- Implementations of these models, using the tools described in Section 5.
- Experimental results for the en-ptBR language pair.

Technical reports will be written during this thesis progress and made publicly available. Paper submission showing intermediate and final results is also planned.

Acknowledgments

This research is supported by FAPESP (Project 2010/03807-4).

References

Eckhard Bick. 2000. *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework.* Ph.D. thesis, Aarhus University.

Helena De Medeiros Caseli and Israel Aono Nunes. 2009. Tradução Automática Estatística baseada em Frases e Fatorada : Experimentos com os idiomas Português do Brasil e Inglês usando o toolkit Moses.

Hubert Comon, Max Dauchet, Remi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. 2007. *Tree automata techniques and applications*, volume 10. Available on: http://www.grappa.univ-lille3.fr/tata.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 128–132.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. Whats in a translation rule? In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL 2004)*, pages 273–280.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*, pages 961–968.

[3] http://www.fjoch.com/GIZA++.html
[4] http://www.isi.edu/licensed-sw/tiburon
[5] http://www.statmt.org/moses

Daniel Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 80–87.

Jonathan Graehl, Kevin Knight, and Jonathan May. 2008. Training Tree Transducers. *Computational Linguistics*, 34:391–427.

Yvette Graham and Josef van Genabith. 2010. Deep Syntax Language Models and Statistical Machine Translation. In *SSST-4 - 4th Workshop on Syntax and Structure in Statistical Translation at COLING 2010*, page 118.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*, pages 609–616.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3):1–49.

Andreas Maletti, Jonathan Graehl, Mark Hopkins, and Kevin Knight. 2009. The power of extended top-down tree transducers. *SIAM Journal on Computing*, 39(2):410–430.

Andreas Maletti. 2010. A Tree Transducer Model for Synchronous Tree-Adjoining Grammars. *Computational Linguistics*, pages 1067–1076.

Jonathan May and Kevin Knight. 2006. Tiburon : A Weighted Tree Automata Toolkit. *Grammars*.

Thai Phuong Nguyen, Akira Shimazu, Tu-Bao Ho, Minh Le Nguyen, and Vinh Van Nguyen. 2008. A tree-to-string phrase-based model for statistical machine translation. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning - CoNLL '08*, pages 143–150.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 295.

Franz Josef Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *HLT-NAACL*, pages 404–411.

Matt Post and Daniel Gildea. 2009. Language modeling with tree substitution grammars. *Computing*, pages 1–8.

Stuart M Shieber. 2004. Synchronous Grammars as Tree Transducers. *Applied Sciences*, pages 88–95.

Vincent Vandeghinste. 2009. Tree-based target language modeling. In *Proceedings of EAMT*, pages 152–159.

Kenji Yamada and Kevin Knight. 2001. A Syntax-based Statistical Translation Model. In *ACL '01 Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530.

Min Zhang, Hongfei Jiang, Aiti Aw, Haizhou Li, Chew Lim Tan, and Sheng Li. 2008. A tree sequence alignment-based tree-to-tree translation model. In *Proc. ACL-08: HLT*, pages 559–567.

ConsentCanvas: Automatic Texturing for Improved Readability in End-User License Agreements

Oliver Schneider & Alex Garnett
Department of Computer Science, University of British Columbia
201-2366 Main Mall, Vancouver, BC, Canada, V6T 1Z4
oschneid@cs.ubc.ca, axfelix@gmail.com

Abstract

We present ConsentCanvas, a system which structures and "texturizes" End-User License Agreement (EULA) documents to be more readable. The system aims to help users better understand the terms under which they are providing their informed consent. ConsentCanvas receives unstructured text documents as input and uses unsupervised natural language processing methods to embellish the source document using a linked stylesheet. Unlike similar usable security projects which employ summarization techniques, our system preserves the contents of the source document, minimizing the cognitive and legal burden for both the end user and the licensor. Our system does not require a corpus for training.

1 Introduction

Less than 2% of users read End-User License Agreement (EULA) documents when indicating their consent to the software installation process (Good et al., 2007). While these documents often serve as a user's sole direct interaction with the legal terms of the software, they are usually not read, as they are presented in such a way as is divorced from the use of the software itself (Friedman et al., 2005). To address this, Kay and Terry (2010) developed what they call *Textured Consent* agreements which employ a linked stylesheet to augment salient parts of a EULA document. Unlike summarization-driven approaches to usable security, this is achieved without any modification of the underlying text, minimizing the cognitive and legal burden for both the end user and the licensor and removing the need to make available a supplementary unmodified document (Kelley et al, 2009; Farzindar, 2004).

We have developed a system, ConsentCanvas, for automating the creation of a Textured Consent document from an unstructured EULA based on the example XHTML/CSS template provided by Kay and Terry (2010; Figure 1). Our system does not currently use any complex syntactic or semantic information from the source document. Instead, it makes use of regular expressions and correlation functions to identify variable-length relevant phrases (Kim and Chan, 2004) to alter the document's structure and appearance.

We report on ConsentCanvas as a work in progress. The system automates the labour intensive manual process used by Kay and Terry (2010). ConsentCanvas has a working implementation, but has not yet been formally evaluated. We also present the first available implementation of Kim and Chan's algorithm (2004).

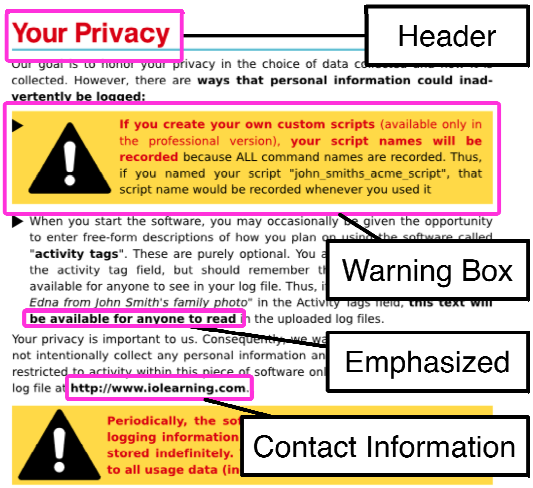

Figure 1. Example Textured Consent Document as designed by Kay and Terry (2010).

Proceedings of the ACL-HLT 2011 Student Session, pages 41–45,
Portland, OR, USA 19-24 June 2011.

2 Methods

We built ConsentCanvas in Python 2.6 using the Natural Language Toolkit (NLTK) 2.0b9. It uses a modified version of the markup.py library available from http://markup.sourceforge.net to generate valid HTML5 documents. A detailed specification of our system workflow is provided in Figure 2. ConsentCanvas was designed with modularity as a priority in order to adapt to the needs of future experimentation and improvement. As such, we contribute not just a working application, but also an extensible framework for the visual embellishment of plaintext documents.

2.1 Analysis

Our system takes plain-text EULA documents as input through a simple command line interface. It then passes this document to four independent submodules for analysis. Each submodule stores the initial and final character positions of a string selected from within the document body, but does not modify the document before reaching the renderer step. This allows for easy extensibility of the system

2.2 Variable-Length Phrase Finder

The variable-length phrase finder module features a Python implementation of the Variable-Length Phrase Finding (VLPF) Algorithm by Kim and Chan (2004). Kim and Chan's algorithm was chosen for its domain independence and adaptability, as it can be fine-tuned to use different correlation functions.

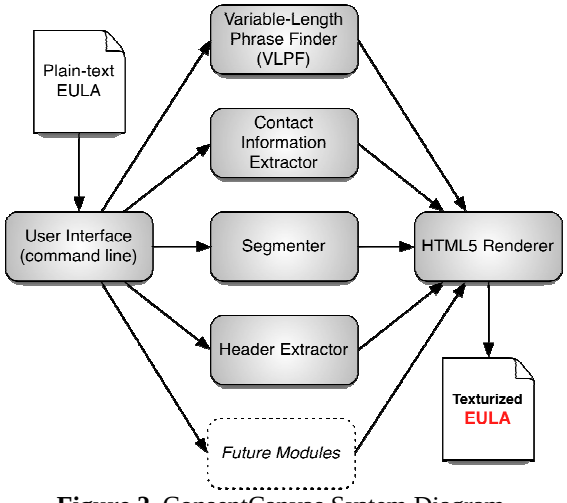

Figure 2. ConsentCanvas System Diagram.

This algorithm computes the conditional probability for the relative importance of variable-length n-gram phrases from the source document alone. It begins by considering every word a phrase with a length of one. The algorithm iteratively increases the length of phrases, adding an adjacent word to the end. That is, every phrase of length m P{m} is considered as P{m-1}w, where w is a following adjacent word.

Correlation is calculated between the leading phrase P{m-1} and the trailing word w. Phrases that maintain a high level of correlation are creating by appending the trailing word w, and those with a correlation score below a certain threshold are pruned before the next iteration. This continues until no more phrases can be created. This method is completely unsupervised.

The VLPF algorithm is able to use any of several existing correlation functions. We have implemented the Piatetsky-Shapiro correlation function, the simplest of the three best-performing functions used by Kim and Chan, which achieved a correlation of 92.0% with human rankings of meaningful phrases (2004).

We removed English stopwords, but did not perform any stemming when selecting relevant phrases because the selection of VLPs did not depend on global term co-occurrence, and we did not want to modify selected exact phrases. We emphasize the top 15% meaningful phrases (as determined by the algorithm) for the entire document. 15% was chosen for its comparable results to Kay and Terry's example document (2010). The phrase selected as the most relevant is also reproduced in the pull quote at the top of the document, as shown in Figure 3.

2.3 Contact Information Extractor

The contact information extractor module uses regular expressions to match URLs, email addresses, or phone numbers within the document text. This information was displayed as bold type in accordance with the Textured Consent template.

2.4 Segmenter

The segmenter module uses Hearst's TextTiling algorithm to "segment text into multi-paragraph subtopic passages" (1997). This algorithm analyzes

patterns of lexical co-occurrence and distribution in order to impose topic boundaries on a document. ConsentCanvas uses the NLTK implementation of the TextTiling algorithm. Segmentation was not applied to the entire document (doing this resulted in a messy layout incoherent with structuring applied by headers and titles). Instead, we used it to identify the lead paragraph of the document, which was rendered differently using the "lead paragraph" container in the template. Future versions will use a more modern segmenting algorithm.

2.5 Header Extractor

The header extractor module uses regular expressions to match any section header-like text from the original document. Several different search strings were used to catch multiple potential header types, including but not limited to:

- 8 OR FEWER ALL-CAPS TOKENS
- 3. Single level numbered headers
- 3.1 Multi-level numbered headers
- Eight or fewer tokens separated by a line break

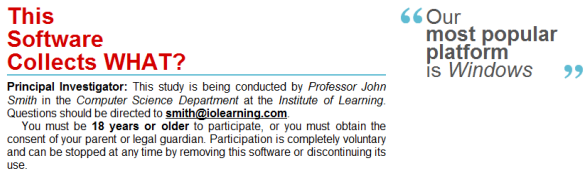

Figure 3. Summary text in the example document.

2.6 Rendering

Each analysis submodule produces a list of character positions where found items begin and end. These are passed to our rendering system, which inserts the corresponding HTML5 tags at the positions in original plaintext EULA. We append a header to the output document to include the linked stylesheet per HTML5 specifications.

3 Analysis & Results

We conducted a brief qualitative analysis on ConsentCanvas after implementation and debugging. However, the problem space and system are not yet ready for formal verification or experimentation. More exploration and refinement are required before we will be able to empirically determine if we have improved readability and comprehension.

3.1 Corpus

We conducted our analysis on a small sample of EULAs from the same collection used by Lavesson et al. (2008) in their work on the classification of EULAs. There were 1021 EULAs in this corpus divided into 96 "bad" and 925 "good" examples. We used the "good" examples for our analysis.

3.2 Variable-Length Phrase Finding Results

Variable-Length Phrases (VLPs) were reasonably effective. In several of the best examples of texturized EULAs security concerns were highlighted; in the texturized version of one document, the pull quote was "on media, ICONIX, Inc. warrants that such media is free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of purchase as evidenced by a copy of the receipt. ICONIX, Inc. warrants." In the same EULA, other VLPs proved helpful: "e that ICONIX, Inc. is free to use any ideas, concepts," "(except one copy for backup purposes)," and "Inc. ICONIX, Inc. does not collect any personally identifiable information regarding senders." Some phrases have incomplete words at the beginning and end; this is an artifact of a known but unfixed bug in the implementation, not a result of the algorithm.

However, these results were mixed in other EULAs. Several short but frequent phrases were found to be VLPs, such as "Inc.," in the same EULA. In short licenses consisting of only one to three paragraphs, sometimes no relevant VLPs were discovered. There are also many phrases that should be highlighted that are not.

3.3 Preliminary System Evaluation

We conducted an informal evaluation in which our system applied texture to 15 documents chosen from our corpus at random. Of these, five were determined to be highly readable exemplar documents. An excerpt from one of these is shown in Figure 4. Of the remaining ten documents, four had poorly selected header markup but were otherwise satisfactory, two were too short or poorly-structured to benefit from the insertion of header markup, two did not perform well on the VLPF step, and two had several errors which appeared to have been caused by the use of non-ASCII characters in the original document.

The pull quote text was nearly unintelligible in almost all cases, due largely to the fact that it did not split evenly on sentence borders. We did not let this detract from our evaluation of the documents, because performance in this area was so consistently, and charmingly, poor, but did not affect readability of the main document body.

4 Discussion

Our preliminary analysis has provided several insights into the challenges and next steps in accomplishing this task.

4.1 Comparisons with Kay and Terry

Kay and Terry (2010) make reference to "augmenting and embellishing" the document text – specifically *not* altering the original content. However, their example document is written concisely in a user-friendly voice dissimilar to most formal EULAs found in the wild. Their work provides a strong proof of concept, but a key line of investigation will be whether their approach is practical, or whether some preprocessing is necessary to simplify content.

4.2 Handling Legal Language

We had anticipated a considerable amount of difficulty in selecting meaningful phrases from difficult-to-understand legal language in the source document. However, most documents were found to contain a number of high-frequency VLPs with both layperson-salient legal terminology and common clues to document structure.

4.3 Future Work

ConsentCanvas is fully implemented but offers many opportunities for improvement as the task becomes better understood. The variable-length phrase finding module only incorporates a single correlation function. More will be added, drawing in particular from those documented by Kim and Chan (2004). Machine learning techniques might also be used to classify phrases as relevant or not, leading to better-emphasized content.

The *rhythm* of emphasized phrasing is also important. In the example license designed by Kay and Terry (2010), there are one or two emphasized phrases in each section. The phrases found by ConsentCanvas are often sporadic, clustering in some sections and absent from others. As a result of this, readability suffers, and so we may need to look into possible stratification of VLPs. This might also aid multi-lingual documents, of which there are a few examples (a cursory look showed the results in French were comparable to those in English in a bilingual EULA in our corpus).

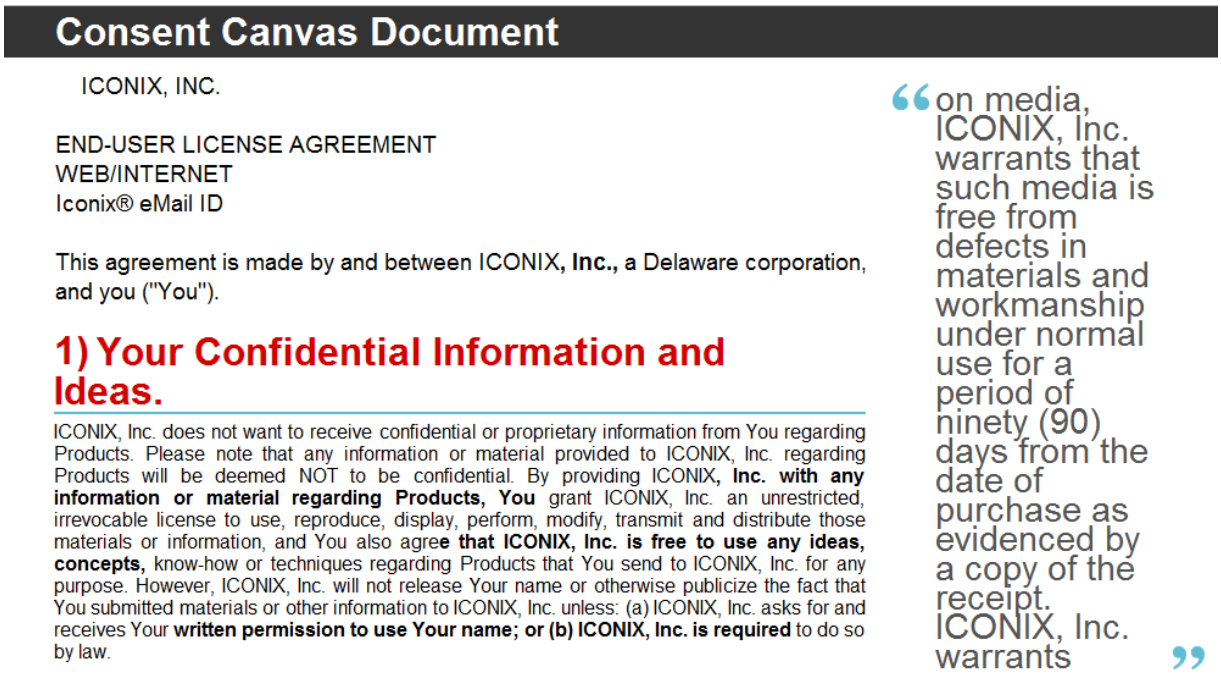

Figure 4. Summary text in an example output document.

Contact information is currently emphasized in the same manner as salient phrases. We plan to eventually embed hyperlinks for all URLs and email addresses found in the source document, as in Kay and Terry (2010).

The segmenter module uses the basic TextTiling algorithm with default parameters. More recent approaches could be implemented and could act on more than the lead paragraph. For example, coherent sections of long EULAs might be identified and presented as separate containers.

We plan to improve header extractor providing more sophisticated regular expressions; we found that a wide variety of header styles were used. In particular, we plan to consider layouts that use digits, punctuation, or inconsistent capitalization in multiple instances in the document body.

There is currently no module that incorporates the "Warning" box from Kay and Terry (2010). This module would be designed to select relevant multiline blocks of text by using techniques similar to the variable-length phrase finder or the segmenter.

ConsentCanvas will also be extended to support command-line parameters. This will enable customized texturing of EULAs and facilitate experimentation for understanding and evaluating gains in comprehension and readability. Finally, we will conduct a formal user evaluation of ConsentCanvas.

5 Conclusion

We have provided a description of the work in progress for ConsentCanvas, a system for automatically adding texture to EULAs to improve readability and comprehension. Informal analysis revealed several key challenges in accomplishing this task and identified the next steps towards exploring effective solutions to this problem.

Acknowledgments

We would like to thank the reviewers for their helpful feedback and Dr. Giuseppe Carenini for his support and encouragement. This work was partially supported by an NSERC CGS M scholarship.

Appendix

The source code, our corpus, and a sample of converted documents are all available at: https://github.com/axfelix/consentCanvas.

References

Farzindar, A. 2004. Legal text summarization by exploration of the thematic structures and argumentative roles. *Text Summarization Branches Out.*

Friedman, B. 2005. Informed consent by design. In *Security and Usability*, Eds. Lorrie Faith Cranor & Simson Garfinkel,

Good, N., Dhamija, R., Grossklags, J., Thaw, D., Aronowitz, S., Mulligan, D. and Konstan, J. 2005. Stopping spyware at the gate: a user study of privacy, notice and spyware. *Proceedings of the 1st Symposium on Usable Privacy and Security*. 43–52.

Hearst, M.A. 1997. TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational linguistics 23*, 1: 33–64.

Kay, M. and Terry, M. 2010. Textured agreements: Re-envisioning electronic consent. *Proceedings of the Sixth Symposium on Usable Privacy and Security.*

Kelley, P.G., Bresee, J., Cranor, L.F., and Reeder, R.W. 2009. A nutrition label for privacy. *Proceedings of the 5th Symposium on Usable Privacy and Security*: 1–12.

Kim, H. and Chan, P.K. 2004. Identifying variable-length meaningful phrases with correlation functions. *16th IEEE International Conference on Tools with Artificial Intelligence*, 30-38.

Lavesson, N., Davidsson, P., Boldt, M., Jacobsson, A. 2008. Spyware Prevention by Classifying End User License Agreements. *Studies in Computational Intelligence, volume 134.* 373-382.

Disambiguating Temporal–Contrastive Discourse Connectives for Machine Translation

Thomas Meyer
Idiap Research Institute / Martigny, Switzerland
EPFL - EDEE doctoral school / Lausanne, Switzerland
Thomas.Meyer@idiap.ch

Abstract

Temporal–contrastive discourse connectives (*although*, *while*, *since*, etc.) signal various types of relations between clauses such as *temporal, contrast, concession* and *cause*. They are often ambiguous and therefore difficult to translate from one language to another. We discuss several new and translation-oriented experiments for the disambiguation of a specific subset of discourse connectives in order to correct some of the translation errors made by current statistical machine translation systems.

1 Introduction

The probabilistic phrase-based models used in statistical machine translation (SMT) have been improved by integrating linguistic information during training stages. Recent attempts include, for example, the reordering of the source language syntax in order to align it closer to the target language word order (Collins et al., 2010) or the tagging of pronouns for grammatical gender agreement (Le Nagard and Koehn, 2010). On the other hand, integrating discourse information, such as discourse relations holding between two spans of text or between sentences, has not yet been applied to SMT.

This paper describes several disambiguation and translation experiments for a specific subset of discourse connectives. Based on examinations in multilingual corpora, we identified the connectives *although, but, however, meanwhile, since, though, when* and *while* as being particularly problematic for machine translation. These discourse connectives signal various types of relations between clauses, such as *temporal, contrast, concession, expansion, cause* and *condition*, which are, as we also show, hard to annotate even by humans. Disambiguating these senses and tagging them in large corpora is hypothesized to help in improving SMT systems to avoid translation errors.

The paper is organized as follows. Section 2 exemplifies translation and human annotation difficulties. Resources and the state of the art for discourse connective disambiguation and parsing are described in Section 3. Section 4 summarizes our experiments for disambiguating the senses of temporal–contrastive connectives. The impact of connective disambiguation on SMT is briefly presented in Section 5. Section 6 concludes the paper with an outline of future work.

2 Translating Connectives

Discourse connectives can signal multiple senses (Miltsakaki et al., 2005). For instance, the connective *since* can have a *temporal* and *causal* meaning. The disambiguation of these senses is crucial to the correct translation of texts from one language to another. Translation can be difficult because there may be no direct lexical correspondence for the explicit source language connective in the target language, as shown by the reference translation of the first example in Table 1, taken from the Europarl corpus (Koehn, 2005).

More often, the incorrect rendering of the sense of a connective can lead to wrong translations, as in the second, third and fourth example in Table 1, which were translated by the Moses SMT decoder (Koehn

Proceedings of the ACL-HLT 2011 Student Session, pages 46–51,
Portland, OR, USA 19-24 June 2011.

EN	*So what we want the European Patent Office to do is something on behalf of the European Commission* ***[while]_temporal*** *the Office itself is not a Community institution.*
FR	*Aussi, ce que nous souhaitons, c'est que l'Office européen des brevets agisse au nom de la Commission européenne* ***[tout en*** *n'été****ant]_temporal*** *pas une institution communautaire.*
EN	*Finally, and in conclusion, Mr President, with the expiry of the ECSC Treaty, the regulations will have to be reviewed* ***[since]_causal*** *I think that the aid system will have to continue beyond 2002...*
FR	**Enfin, et en conclusion, Monsieur le président, à l'expiration du traité ceca, la réglementation devra être revu* ***[depuis que]_temporal*** *je pense que le système d'aides devront continuer au-delà de 2002...*
EN	*Between 1998 and 1999, loyalists assaulted and shot 123 people,* ***[while]_contrast*** *republicans assaulted and shot 93 people.*
FR	*Entre 1998 et 1999, les loyalistes ont attaqué et abattu 123 personnes, [___] 93 pour les républicains.*
EN	*He said Akzo is considering alliances with American drug companies,* ***[although]_contrast*** *he wouldn't elaborate.*
DE	**Er sagte Akzo erwägt Allianzen mit amerikanischen Pharmakonzerne,* ***[obwohl]_concession*** *er möchte nicht näher eingehen.*

Table 1: Translation examples from Europarl and the PDTB. The discourse connectives, their translations, and their senses are indicated in bold. The first example is a reference translation from EN into FR, while the second, third and fourth example are wrong translations generated by MT (EN–FR and EN–DE), hence marked with an asterisk.

et al., 2007) trained on the Europarl EN–FR and respectively EN–DE subcorpora. The reference translation for the second example uses the French connective *car* with a correct *causal* sense, instead of the wrong *depuis que* generated by SMT, which expresses a *temporal* relation. In the third example, the SMT system failed to translate the English connective *while* to French. The French translation is therefore not coherent, the *contrastive* discourse information cannot be established without an explicit connective. The last example in Table 1 is a sentence from the Penn Discourse Treebank (Prasad et al., 2008), see Section 3. In its German translation, it would be correct to use the connective *auch wenn* (for *contrast*) instead of *obwohl* (for *concession*).

These examples illustrate the difficulties in translating discourse connectives, even when they are lexically explicit. Our hypothesis is, that the automatic annotation of the senses prior to translation can help finding more often the correct lexical correspondences of a connective (see Section 5 for one of the methods to achieve this).

while (489)	**Translation EN-FR**
56% T	tout en V-gerund (22%), tant que (22%), tandis que (11%)
30% CT	tandis que (56%), alors que (40%)
14% CO	même si (100%)
although (347)	**Translation EN-DE**
76.7% CO	obwohl (74%), zwar (9%), auch wenn (9%)
23.3% CT	obgleich (43%), obwohl (29%)

Table 2: The English connectives *while* and *although* in the Europarl corpus (sections numbered 199x, EN-FR and EN-DE) with token frequency, sense distribution and most frequent translations ordered by the corresponding senses (T = *temporal*, CO = *concession*, CT = *contrast*).

When examining the frequency and sense distribution of these connectives and their translations in the Europarl corpus, the results confirm that at least such a fine-grained disambiguation as the one between *contrast* and *concession* is necessary for a correct translation. Table 2 shows cases where the different senses of the connectives *while* and *although* lead to different translations. Disambiguation of the senses here can help finding the correct lexical correspondence of the connective.

To confirm that the automatic translation of discourse connectives is not straightforward, we annotated 80 sentences from the Europarl corpus containing the connective *while* with the corresponding sense (T, CO or CT) and another 60 sentences containing the French connective *alors que* (T or CT). We then translated these sentences with the already mentioned EN–FR and FR–EN Moses SMT system and compared the output manually to the reference translations from the corpus. The overall system performance was 61% of correct translations for sentences with *while* and 55% of correct translations with *alors que*. As mistakes we either counted missing target connective words (only when the output sentence became incoherent) or wrong connective words because of failure in correct sense rendering.

Also, the *manual* sense annotation task is not trivial. In a manual annotation experiment, the senses of the connective *while* (T, CO and CT) were indicated in 30 sentences by 4 annotators. The overall agreement on the senses was not higher than a kappa value of 0.6, which is acceptable but would need improvement in order to produce a reliable resource.

3 Data and Related Work

One of the few available discourse annotated corpora in English is the Penn Discourse Treebank (PDTB) (Prasad et al., 2008). For this resource, one hundred types of explicit connectives were manually annotated, as well as implicit relations not signaled by a connective.

For French, the ANNODIS project for annotation of discourse (Pery-Woodley et al., 2009) will provide an original, discourse-annotated corpus. Resources for Czech are also becoming available (Zikanova et al., 2010). For German, a lexicon of discourse connectives exists since the 1990s, namely DiMLex for lexicon of discourse markers (Stede and Umbach, 1998). An equivalent, more recent database for French is LexConn for lexicon of connectives (Roze et al., 2010) – containing a list of 328 explicit connectives. For each of them, LexConn indicates and exemplifies the possible senses, chosen from a list of 30 labels inspired from Rhetorical Structure Theory (Mann and Thompson, 1988).

For the first classification experiments in Section 4, we concentrated on English and the explicit connectives in the PDTB data. The sense hierarchy used in the PDTB consists of three levels, reaching from four top level senses (*Temporal, Contingency, Comparison* and *Expansion*) via 16 subsenses on the second level to 23 further subsenses on the third level. As the annotators were allowed to assign one or two senses for each connective there are 129 possible simple or complex senses for more than 18,000 explicit connectives. The PDTB further sees connectives as discourse-level predicates that have two propositional arguments. Argument 2 is the one containing the explicit connective. The sentence from the first example in Table 1 can be represented as *while(So what we...*[argument 1], *the Office itself...*[argument 2]), which is very helpful to examine the context of a connective (see Section 4.1 on features).

The release of the PDTB had quite an impact on disambiguation experiments. The state of the art for recognizing explicit connectives in English is therefore already high, at a level of 94% for disambiguating the four main senses on the first level of the PDTB sense hierarchy (Pitler and Nenkova, 2009). However, when using all 100 types of connectives and the whole PDTB training set, it is not so difficult to achieve such a high score, because of the large amount of instances and the rather broad distinction of the four main classes only. As we show in the next section, when building separate classifiers for specific connectives with senses from the more detailed second hierarchy level of the PDTB, it is more difficult to reach high accuracies. Recently, Lin et al. (2010) built the first end-to-end PDTB discourse parser, which is able to parse unrestricted text with an F1 score of 38.18% on PDTB test data and for senses on the second hierarchy level.

4 Disambiguation Experiments

For the experiments described here we used the WEKA machine learning toolkit (Hall et al., 2009) and its implementation of a RandomForest classifier (Breiman, 2001). This method outperformed, in our task, the C4.5 decision tree and NaiveBayes algorithms often used in recent research on discourse connective classification.

Our first experiment was aimed at sense disambiguation down to the third level of the PDTB hierarchy. The training set here consisted of all 100 types of explicit connectives annotated in the PDTB training set (15,366 instances). To make the figures and results of this paper comparable to related work, we use the subdivision of the PDTB recommended in the annotation manual: sections 02–21 as training set and section 23 as test set. The only two features were the (capitalized) connective word tokens from the PDTB and their Part of Speech (POS) tags. For *all 129 possible sense combinations*, including complex senses, results reach *66.51% accuracy* with 10-fold cross validation on the training set and *74.53% accuracy* on the PDTB test set[1]. This can be seen as a baseline experiment. For instance, Pitler and Nenkova (2009) report an accuracy of 85.86% for correctly classified connectives (with the 4 main senses), when using the connective token as the only feature.

Based on the analysis of translations and frequencies from Section 2, we then reduced the list of senses to the following six: *temporal* (T), *cause* (C),

[1] As far as we know, Versley (2010) is the only reference reporting results down to the third level, reaching an accuracy of 79%, using more features, but not stating whether the complex sense annotations were included.

Connective	Senses with number of occurrences	Best feature subset	Accuracy	Baseline	kappa
although	134 CO, 133 CT	8, 9, 10	**58.4%**	48.7%	0.17
but	2090 CT, 485 CO, 77 E	5, 8, 9, 10	76.4%	78.8%	0.02
however	261 CT, 119 CO	1–10	68.4%	68.7%	0.05
meanwhile	77 T, 57 E, 22 CT	1–10	51.9%	49.4%	0.09
since	83 C, 67 T	1, 4, 6, 8, 9, 10	**75.3%**	55.3%	0.49
though	136 CO, 125 CT	1, 2, 3, 9, 10	**65.1%**	52.1%	0.30
when	640 T, 135 COND, 17 C, 8 CO, 2 CT	1, 2, 10	79.9%	79.8%	0.05
while	342 CT, 159 T, 77 CO, 53 E	3, 5, 7, 8, 9, 10	**59.6%**	54.1%	0.23
all	2975 CT, 959 CO, 943 T, 187 E, 135 COND, 100 C	1–10	**72.6%**	56.1%	0.50

Table 3: Disambiguation of temporal–contrastive connectives.

condition (COND), *contrast* (CT), *concession* (CO) and *expansion* (E). All subsenses from the third PDTB hierarchy level were merged under second level ones (C, COND, CT, CO). Exceptions were the top level senses T and E, which, so far, need no further disambiguation for translation. In addition, we extracted separate training sets for each of the 8 temporal–contrastive connectives in question and one training set for all them. The number of occurrences and senses in the sets for the single connectives is listed in Table 3. The total number of instances in the training set for all 8 connectives is 5,299 occurrences, with a sense distribution of 56.1% CT, 18% CO, 17.8% T, 3.5% E, 2.5% COND, 1.9% C.

Before summarizing the results, we describe the features implemented and used so far.

4.1 Features

The following basic surface features were considered when disambiguating the senses signaled by connectives. Their values were extracted from the PDTB manual gold annotation. Future automated disambiguation will be applied to unrestricted text, identifying the discourse arguments and syntactical elements in automatically parsed and POS–tagged sentences.

1. the (capitalized) connective word form
2. its POS tag
3. first word of argument 1
4. last word of argument 1
5. first word of argument 2
6. last word of argument 2
7. POS tag of the first word of argument 2
8. type of first word of argument 2
9. parent syntactical categories of the connective
10. punctuation pattern

The cased word forms (feature 1) were left as is, therefore also indicating whether the connective is located at the beginning of a sentence or not. The variations from the PDTB (e.g. *when – back when* etc.) were also included, supplemented by their POS tags (feature 2). As shown by Lin et al. (2010) and duVerle and Prendinger (2009), the context of a connective is very important. The arguments may include other (reinforcing or opposite) connectives, numbers and antonyms (to express contrastive relations). We extracted the words at the beginning and at the end of argument 1 (features 3, 4) and argument 2 (features 5, 6) which are, as observed, other connectives, gerunds, adverbs or determiners (further generalized by features 7 and 8). The paths to syntactical ancestors (feature 9) in which the connective word form appears are quite numerous and were therefore truncated to a maximum of four ancestors (e.g. |SBAR‖VP‖S|, |ADVP‖ADJP‖VP‖S|, etc). Punctuation patterns (feature 10) are of the form C,A – A,CA etc. where C is the explicit connective and A a placeholder for all the other words. Punctuation is important for locating connectives as many of them are subordinating and coordinating conjunctions, separated by commas (Haddow, 2005, p. 23).

4.2 Results

In the disambiguation experiments described here, results were generated separately for every temporal–contrastive connective (supposing one may try to improve the translation of only certain connectives), in addition to one result for the whole subset. The results in Table 3 above are based on 10-fold cross validation on the training sets. They were measured using accuracy (percentage of correctly classified instances) and the kappa

value. The baseline is the majority class, i.e. the prediction for the most frequent sense annotated for the corresponding connective. Feature selection was performed in order to find the best feature subset, which also improved the accuracy in a range of 1% to 2%. Marked in bold are the accuracy values significantly above the baseline ones[2]. The last result for all 8 temporal–contrastive connectives reports a six-way classification of senses very close to one another: the accuracy and kappa values are well above random agreement and prediction of the majority class.

Note that experiments for specific subsets of connectives have very rarely been tried in research. Miltsakaki et al. (2005) describe results for *since, while* and *when*, reporting accuracies of 89.5%, 71.8% and 61.6%. The results for the single connectives are comparable with ours in the case of *since* and *while*, where similar senses were used. For *when* they only distinguished three senses, whereas we report a higher accuracy for 5 different senses, see Table 3.

5 SMT Experiments

We have started to explore how to constrain an SMT system to use labeled connectives resulting from the experiments above. There are at least two methods to integrate labeled discourse connectives in the SMT process. A first method modifies the phrase table of the Moses SMT decoder (Koehn et al., 2007) in order to encourage it to translate a specific sense of a connective with an acceptable equivalent. A second, more natural method for an SMT system would be to apply the discourse information obtained from the disambiguation module, adding the sense tags to the discourse connectives in a large parallel corpus. This corpus could then be used to train a new SMT system learning and weighting these tags during the training.

So far, we experimented with method one. Information about the possible senses of the connective *while*, labeled as *temporal*(1), *contrast*(2) or *concession*(3)) was directly introduced to the English source language phrases when there was an appropriate translation of the connective in the French equivalent phrase. We also increased the lexical probability scores for such modified phrases. The following example gives an idea of the changes in the phrase table of the above-mentioned EN–FR Moses SMT system:

< *original:*
and the commission , while preserving ||| et la commission tout en défendant ||| 1 3.8131e-06 1 5.56907e-06 2.718 ||| ||| 1 1
and while many ||| et bien que de nombreuses ||| 1 0.00140575 0.5 0.000103573 2.718 ||| ||| 1 1

> *modified:*
and the commission , **while-1** preserving ||| et la commission tout en défendant ||| 1 **1** 1 **1** 2.718 ||| ||| 1 1
and **while-3** many ||| et bien que de nombreuses ||| 1 **1** 0.5 **1** 2.718 ||| ||| 1 2

Experiments with such modifications have already demonstrated a slight increase of BLEU scores (by 0.8% absolute) on a small test corpus (20 hand-labeled sentences). The analysis of results has shown that the system behaves as expected, i.e. labeled connectives are correctly translated. This tends to confirm the hypothesis of this paper, that information regarding discourse connectives indeed can lead to better translations.

6 Conclusion and Future Work

The paper described new translation-oriented approaches to the disambiguation of a subset of explicit discourse connectives with highly ambiguous temporal–contrastive senses. Although lexically explicit, their translation by current SMT systems is often wrong. Disambiguation results in reasonably high accuracies but also shows that one should find more accurate and additional features. We will try to better model the context of a connective, for instance by integrating word similarity distances from WordNet as features.

In addition, the paper showed a first method to force an existing and trained SMT system to translate discourse connectives correctly. This led to noticeable improvements on the translations of the tested sentences. We will continue to train SMT systems on automatically labeled discourse connectives in large corpora.

Acknowledgments

This work is funded by the Swiss National Science Foundation (SNSF) under the Project Sinergia

[2]Paired t-tests were performed at 95% confidence level. The other accuracy values are either near to the baseline ones or not significantly below them.

COMTIS, contract number CRSI22_127510, www.idiap.ch/comtis/. Many thanks go to Dr. Andrei Popescu-Belis, Dr. Bruno Cartoni and Dr. Sandrine Zufferey, for insightful comments and collaboration.

References

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

Michael Collins, Phillipp Koehn, Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. *Proceedings of the 43rd Annual Meeting of the ACL*, 531–540

David duVerle, Helmut Prendinger. 2009. A Novel Discourse Parser Based on Support Vector Machine Classification. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 665–673.

Barry Haddow. 2005. Acquiring a Disambiguation Model For Discourse Connectives. *Master Thesis. University of Edinburgh, School of Informatics*.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1).

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *Proceedings of MT Summit X*, 79–86.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbs. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of the 45th Annual Meeting of the ACL, Demonstration session*, 177–180.

Ronan Le Nagard, Philipp Koehn. 2010. Aiding Pronoun Translation with Co-Reference Resolution. *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and Metrics MATR*, 258–267.

Ziheng Lin, Hwee Tou Ng, Min-Yen Kan. 2010. A PDTB-Styled End-to-End Discourse Parser. *Technical Report TRB8/10. School of Computing, National University of Singapore*, 1–15.

William C. Mann, Sandra A. Thompson. 1988. Rhetorical structure theory: towards a functional theory of text organization. *Text* 8(3):243–281.

Eleni Miltsakaki, Nikhil Dinesh, Rashmi Prasad, Aravind Joshi, Bonnie Webber. 2005. Experiments on Sense Annotations and Sense Disambiguation of Discourse Connectives. *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT)*.

Marie-Paule Péry-Woodley, Nicholas Asher, Patrice Enjalbert, Farah Benamara, Myriam Bras, Cécile Fabre, Stéphane Ferrari, Lydia-Mai Ho-Dac, Anne Le Draoulec, Yann Mathet, Philippe Muller, Laurent Prévot, Josette Rebeyrolle, Ludovic Tanguy, Marianne Vergez-Couret, Laure Vieu, Antoine Widlöcher. 2009. ANNODIS: une approche outille de l'annotation de structures discursives. *Proceedings of TALN*.

Emily Pitler, Ani Nenkova. 2009. Using Syntax to Disambiguate Explicit Discourse Connectives in Text. *Proceedings of the ACL-IJCNLP 2009 Conference, Short Papers*. 13–16.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, Bonnie Webber. 2008. The Penn Discourse Treebank 2.0. *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, 29641-2968.

Charlotte Roze, Laurence Danlos, Philippe Muller. 2010. LEXCONN: a French Lexicon of Discourse Connectives. *Proceedings of Multidisciplinary Approaches to Discourse (MAD)*.

Manfred Stede, Carla Umbach. 1998. DiMLex: a lexicon of discourse markers for text generation and understanding. *Proceedings of the 36th Annual Meeting of the ACL*, 1238–1242.

Yannick Versley. 2010. Discovery of Ambiguous and Unambiguous Discourse Connectives via Annotation Projection. *Proceedings of Workshop on Annotation and Exploitation of Parallel Corpora (AEPC)*, 83–82

Sárka Zikánová, Lucie Mladová, Jiří Mírovský, Pavlina Jínová. 2010. Typical Cases of Annotators' Disagreement in Discourse Annotations in Prague Dependency Treebank. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, 2002–2006.

PsychoSentiWordNet

Amitava Das
Department of Computer Science and Engineering
Jadavpur University
amitava.santu@gmail.com

Abstract

Sentiment analysis is one of the hot demanding research areas since last few decades. Although a formidable amount of research has been done but still the existing reported solutions or available systems are far from perfect or to meet the satisfaction level of end user's. The main issue may be there are many conceptual rules that govern sentiment, and there are even more clues (possibly unlimited) that can convey these concepts from realization to verbalization of a human being. Human psychology directly relates to the unrevealed clues; govern the sentiment realization of us. Human psychology relates many things like social psychology, culture, pragmatics and many more endless intelligent aspects of civilization. Proper incorporation of human psychology into computational sentiment knowledge representation may solve the problem. PsychoSentiWordNet is an extension over SentiWordNet that holds human psychological knowledge and sentiment knowledge simultaneously.

1 Introduction

In order to identify sentiment from a text, lexical analysis plays a crucial role. For example, words like ***love***, ***hate***, ***good*** and ***favorite*** directly indicate sentiment or opinion. Various previous works (Pang et al., 2002; Wiebe and Mihalcea, 2006; Baccianella et. al., 2010) have already proposed techniques for making dictionaries for those sentiment words. But polarity assignment of such sentiment lexicons is a hard semantic disambiguation problem. The regulating aspects which govern the lexical level semantic orientation are natural language context (Pang et al., 2002), language properties (Wiebe and Mihalcea, 2006), domain pragmatic knowledge (Aue and Gamon, 2005), time dimension (Read, 2005), colors and culture (Strapparava and Ozbal, 2010) and many more unrevealed hidden aspects. Therefore it is a challenging and enigmatic research problem.

What previous studies proposed is to attach prior polarity to each sentiment lexicon level. Prior polarity is an approximation value based on corpus heuristics based statistics and not exact. The probabilistic fixed point prior polarity scores do not solve the problem completely rather it shoves the problem into next level, called contextual polarity classification.

The hypothesis we started with is that the summation of all the regulating aspects of sentiment orientation is human psychology and thus it is called multi-faceted problem (Liu, 2010). More precisely what we meant by human psychology is the all known and unknown aspects, directly or indirectly govern the sentiment orientation knowledge of us. The regulating aspects wrapped in the present PsychoSentiWordNet are ***Gender***, ***Age***, ***City***, ***Country***, ***Language*** and ***Profession***.

The PsychoSentiWordNet is an extension over the existing SentiWordNet to hold the possible psychological ingredients, governs the sentiment understandability of us. The PsychoSentiWordNet holds variable prior polarity scores, could be fetched depending upon those psychological regulating aspects. An example may illustrate the definition better for the concept "***Rock_Climbing***":

Aspects (Age)	Polarity
Null	Positive
50-54	Negative
26-29	Positive

Proceedings of the ACL-HLT 2011 Student Session, pages 52–57,
Portland, OR, USA 19-24 June 2011.

In the previous example the described concept "***Rock_Climbing***" is generally positive as it is adventurous and people have it to make fun or excursion. But it demands highly physical ability thus may be not as good for aged people like the younger people.

PsychoSentiWordNet provides good coverage as it an extension over SentiWordNet 3.0 (Baccianella et. al., 2010). In this paper, we propose an interactive gaming (Dr Sentiment) technology to collect psycho-sentimental polarity for lexicons.

In this section we have philosophically argued about the necessity of developing PsychoSentiWordNet. In the next section we will describe about the technical proposed architecture for building the lexical resource. Section 3 explains about some exciting outcomes that support the usefulness of the PsychoSentiWordNet. What we believe is the developed PsychoSentiWordNet will help automatic sentiment analysis research in many aspect and other disciplines as well, described in the section 4.The data structure and organization is described in section 5 and finally the present paper concluded with section 6.

2 Dr Sentiment

Dr Sentiment [1] is a template based interactive online game, which collects player's sentiment by asking a set of simple template based questions and finally reveals a player's sentimental status. Dr Sentiment fetches random words from SentiWordNet synsets and asks every player to tell about his/her sentiment polarity understanding regarding the concept behind.

There are several motivations behind developing an intuitive game to automatically collect human psycho-sentimental orientation information.

In the history of Information Retrieval research there is a milestone when ESP game[2] (Ahn et al., 2004) innovate the concept of a game to automatically label images available in the World Wide Web. It has been identified as the most reliable strategy to automatically annotate the online images. We are highly motivated by the success of the Image Labeler game.

A number of research endeavors could be found in literature for creation of Sentiment Lexicon in several languages and domains. These techniques can be broadly categorized in two genres, one follows classical manual annotation (Andreevskaia and Bergler, 2006);(Wiebe and Riloff, 2006); (Mohammad et al., 2008) techniques and the others proposed various automatic techniques (Tong, 2001). Both types of techniques have few limitations. Manual annotation techniques are undoubtedly trustable but it generally takes time. Automatic techniques demands manual validations and are dependent on the corpus availability in the respective domain. Manual annotation technique required a large number of annotators to balance one's sentimentality in order to reach agreement. But human annotators are quite unavailable and costly.

But sentiment is a property of human intelligence and is not entirely based on the features of a language. Thus people's involvement is required to capture the sentiment of the human society. We have developed an online game to attract internet population for the creation of PsychoSentiWordNet automatically. Involvement of Internet population is an effective approach as the population is very high in number and ever growing (approx. 360,985,492) [3] . Internet population consists of people with various languages, cultures, age etc and thus not biased towards any domain, language or particular society. The Sign Up form of the "Dr Sentiment" game asks the player to provide personal information such as Sex, Age, City, Country, Language and Profession.

The lexicons tagged by this system are credible as it is tagged by human beings. In either way it is not like a static sentiment lexicon set as it is updated regularly. Almost 100 players per day are currently playing it throughout the world in different languages.

The game has four types of question templates. For further detailed description the question templates are named as Q1, Q2, Q3 and Q4. To make the gaming interface more interesting images has been added with the help of Google image search API [4] and to avoid biasness we have randomized among the first ten images retrieved by Google. Snapshots of different screens from the game are presented in Figure 1.

[1] http://www.amitavadas.com/Sentiment%20Game/
[2] http://www.espgame.org/
[3] http://www.internetworldstats.com/stats.htm
[4] http://code.google.com/apis/imagesearch/

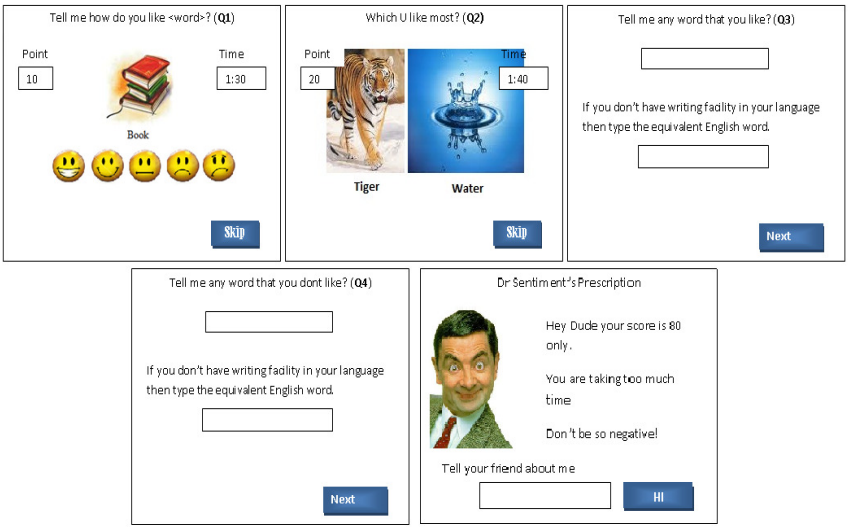

Figure 1: Snapshots from Dr Sentiment Game

2.1 Gaming Strategy

There are four types of questions: Q1, Q2, Q3 and Q4. Dr Sentiment asks 30 questions to each player.There are predefined distributions of each question type as 11 for Q1, 11 for Q2, 4 for Q3 and 4 for Q4. There is no thumb rule behind the cardinals rather they are arbitrarily chosen and randomly changed for experimentation. The questions are randomly asked to keep the game more interesting.

2.2 Q1

An English word from the English SentiWordNet synset is randomly chosen. The Google image search API is fired with the word as a query. An image along with the word itself is shown in the Q1 page of the game.

Players press the different emoticons (Fig 2) to express their sentimentality. The interface keeps log records of each interaction.

Figure 2: Emoticons to Express Player's Sentiment

2.3 Q2

This question type is specially designed for relative scoring technique. For example: ***good*** and ***better*** both are positive but we need to know which one is more positive than other. Table 1 shows how in SentiWordNet relative scoring has been made. With the present gaming technology relative polarity scoring has been assigned to each ***n-n*** word pair combination.

Now about the technical solution how we did it. Randomly ***n*** (presently 2-4) words have been chosen from the source SentiWordNet synsets along with their images as retrieved by Google API. Each player is then asked to select one of them that he/she likes most. The relative score is calculated and stored in the corresponding log log table.

Word	Positivity	Negativity
Good	0.625	0.0
Better	0.875	0.0
Best	0.980	0.0

Table 1: Relative Sentiment Scores from SentiWordNet

2.4 Q3

The player is asked for any positive word in his/her mind. This technique helps to increase the coverage of existing SentiWordNet. The word is then added to the PsychoSentiWordNet and further used in Q1 to other users to note their sentimentality about the particular word.

2.5 Q4

A player is asked by Dr Sentiment about any negative word. The word is then added to the PsychoSentiWordNet and further used in Q1 to

other users to note their sentimentality about the particular word.

2.6 Comment Architecture

There are three types of Comments, Comment type 1 (CMNT1), Comment type 2 (CMNT2) and the final comment as Dr Sentiment's prescription. CMNT1 type and CMNT2 type comments are associated with question types Q1 and Q2 respectively.

2.7 CMNT1

Comment type 1 has 5 variations as shown in the Comment table in Table 3. Comments are randomly retrieved from comment type table according to their category.

- Positive word has been tagged as negative (PN)
- Positive word has been tagged as positive (PP)
- Negative word has been tagged as positive (NP)
- Negative word has been tagged as negative (NN)
- Neutral (NU)

2.8 CMNT2

The strategy here is as same as the CMNT 1. Comment type 2 has only 2 variations as.

- Positive word has been tagged as negative. (PN)
- Negative word has been tagged as positive (NP)

2.9 Dr Sentiment's Prescription

The final prescription depends on various factors such as total number of positive, negative or neutral comments and the total time taken by any player. The final prescription also depends on the range of the values of accumulating all the above factors.

This is only the appealing factor to a player. The provoking message for players is Dr Sentiment can reveal their sentimental status: whether they are extreme negative or positive or very much neutral or diplomatic etc. A word previously tagged by a player is avoided by the tracking system for the next time playing as our intension is to tag more and more words involving Internet population. We observe that the strategy helps to keep the game interesting as a large number of players return to play the game after this strategy was implemented.

We are not demanding that the revealed status of a player by Dr Sentiment is exact or ideal. It is only to make fun but the outcomes of the game effectively help to store human sentimental psychology in terms of computational lexicon.

3 Senti-Mentality

PsychoSentiWordNet gives a good sketch to understand the psycho-sentimental behavior of society depending upon proposed psychological dimensions. The PsychoSentiWordNet is basically the log records of every player's tagged words.

3.1 Concept-Culture-Wise Analysis

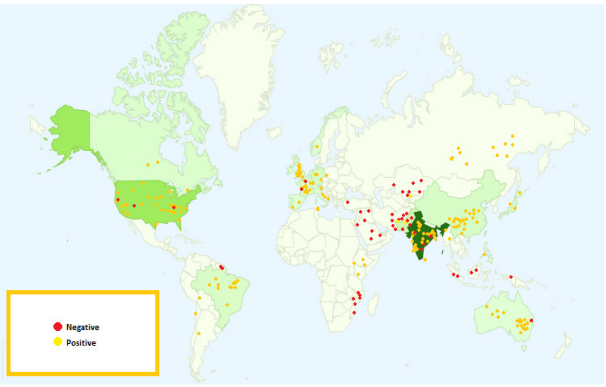

Figure 3: Geospatial Senti-Mentality

The word "***blue***" get tagged by different players around the world. But surprisingly it has been tagged as positive from one part of the world and negative from another part of the world. The graphical illustration in Figure 3 explains the situation. The observation is that most of the negative tags are coming from the middle-east and especially from the Islamic countries. We found a line in Wiki[5] (see in Religion Section) that may give a good explanation: "Blue in Islam: In verse 20:102 of the Qur'an, the word زرق zurq (plural of azraq 'blue') is used metaphorically for evil doers whose eyes are glazed with fear". But other explanations may be there for this. This is an interesting observation that supports the effectiveness of PsychoSentiWordNet. This information could be further retrieved from the developed source by giving information like (blue, Italy), (blue, Iraq) or (blue, USA) etc.

3.2 Age-Wise Analysis

Another interesting observation is that sentimentality may vary age-wise. For better understanding we look at the total statistics and the

[5] http://en.wikipedia.org/wiki/Blue

age wise distribution of all the players. Total 533 players have taken part till date. The total number of players for each range of age is shown at top of every bar. In the Figure 4 the horizontal bars are divided into two colors (Green depicts the Positivity and Red depicts the negativity) according to the total positivity and negativity scores, gathered during playing. This sociological study gives an idea that variation of sentimentality with age. This information could be further retrieved from the developed source by giving information like (X, 36-39) or (X, 45-49) etc.

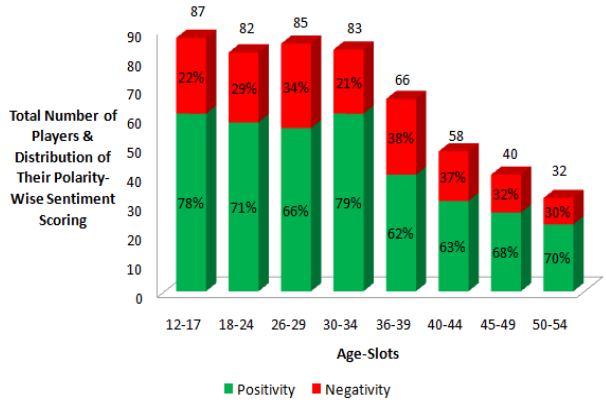

Figure 4: Age-Wise Senti-Mentality

3.3 Gender Specific

It is observed from the statistics collected that women are more positive than a man. The variations in sentimentality among men and women are shown in the following Figure 5.

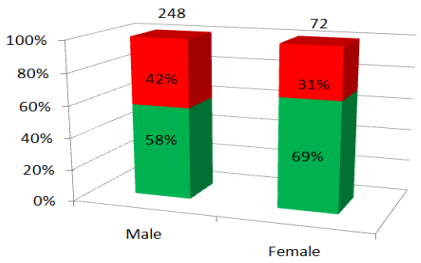

Figure 5: Gender Specific Senti-Mentality

3.4 Other-Wise

We have described several important observations in the previous sections and there are other important observations as well. Studies on the combinations of the proposed psychological dimensions, such as, location-age, location-profession and gender-location may reveal some interesting results.

4 Expected Impact of the Resource

Undoubtedly the generated PsychoSentiWordNet are important resource for sentiment/opinion or emotion analysis task. Moreover the other non linguistic psychological dimensions are very much important for further analysis and in several newly discovered sub-disciplines such as: Geospatial Information retrieval (Egenhofer, 2002), Personalized search (Gaucha et al., 2003) and Recommender System (Adomavicius and Tuzhilin, 2005) etc.

5 The Data Structure and Organization

Deciding about the data structure of this kind of special requirement was not trivial. Presently RDBMS (Relational Database Management System) has been used. Several tables are being used to keep user's clicking log and their personal information.

As one of the research motivations was to generate up-to-date prior polarity scores thus we decided to generate web service API by that people could access latest prior polarity scores. We do believe this method will over perform than a static sentiment lexicon set.

6 Conclusion & Future Direction

In the present paper the development of the PsychoSentiWordNet has been described. No evaluation has been done yet as there is no data available for this kind of experimentation and to the best of our knowledge this is the first endeavor where sentiment meets psychology.

Our present goal is to collect such corpus and experiment to check whether variable prior polarity score of PsychoSentiWordNet excel over the fixed point prior polarity score of SentiWordNet.

Acknowledgments

The work reported in this paper was supported by a grant from the India-Japan Cooperative Program (DST-JST) 2009 Research project entitled "***Sentiment Analysis where AI meets Psychology***" funded by Department of Science and Technology (DST), Government of India.

References

Andreevskaia Alina and Bergler Sabine. CLaC and CLaC-NB: Knowledge-based and corpus-based approaches to sentiment tagging. In the Proc. of the 4th SemEval-2007, Pages 117–120, Prague, June 2007.

Ahn Luis von and Laura Dabbish. Labeling Images with a Computer Game. In the Proc. of ACM-CHI 2004.

Aue A. and Gamon M., Customizing sentiment classifiers to new domains: A case study. In the Proc. of RANLP, 2005.

Baccianella Stefano, Andrea Esuli, and Fabrizio Sebastiani. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In the Proc. of LREC-10.

Bing Liu. Sentiment Analysis: A Multi-Faceted Problem.In the IEEE Intelligent Systems, 2010.

Mohammad Saif, Dorr Bonnie and Hirst Graeme. Computing Word-Pair Antonymy. In the Proc. of EMNLP-2008.

Pang Bo, Lee Lillian, and Vaithyanathan Shivakumar. Thumbs up? Sentiment classification using machine learning techniques. In the Proc. of EMNLP, Pages 79–86, 2002.

Read Jonathon. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In the Proc. of the ACL Student Research Workshop, 2005.

Strapparava, C. and Valitutti, A. WordNet-Affect: an affective extension of WordNet. In Proc. of LREC 2004, Pages 1083 – 1086

Wiebe Janyce and Mihalcea Rada. Word sense and subjectivity. In the Proc. of COLING/ACL-06. Pages 1065-1072.

Wiebe Janyce and Riloff Ellen. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In the Proc. CICLING, Pages 475–486, 2006.

Richard M. Tong. An operational system for detecting and tracking opinions in online discussion. In the Proc. of the Workshop on Operational Text Classification (OTC), 2001.

Optimistic Backtracking
A Backtracking Overlay for Deterministic Incremental Parsing

Gisle Ytrestøl
Department of Informatics
University of Oslo
gisley@ifi.uio.no

Abstract

This paper describes a backtracking strategy for an incremental deterministic transition-based parser for HPSG. The method could theoretically be implemented on any other transition-based parser with some adjustments. In this paper, the algorithm is evaluated on CuteForce, an efficient deterministic shift-reduce HPSG parser. The backtracking strategy may serve to improve existing parsers, or to assess if a deterministic parser would benefit from backtracking as a strategy to improve parsing.

1 Introduction

Incremental deterministic parsing has received increased awareness over the last decade. Processing linguistic data linearly is attractive both from a computational and a cognitive standpoint. While there is a rich research tradition in statistical parsing, the predominant approach derives from chart parsing and is inherently non-deterministic.

A deterministic algorithm will incrementally expand a syntactic/semantic derivation as it reads the input sentence one word/token at the time. There are a number of attractive features to this approach. The time-complexity will be linear when the algorithm is deterministic, i.e. it does not allow for later changes to the partial derivation, only extensions to it. For a number of applications, e.g. speech recognition, the ability to process input on the fly per word, and not per sentence, can also be vital. However, there are inherent challenges to an incremental parsing algorithm. Garden paths are the canonical example of sentences that are typically misinterpret due to an early incorrect grammatical assumption.

(1) The horse raced past the barn fell.

The ability to reevaluate an earlier grammatical assumption is disallowed by a deterministic parser. *Optimistic Backtracking* is an method designed to locate the incorrect parser decision in an earlier stage if the parser reaches an illegal state, i.e. a state in which a valid parse derivation cannot be retrieved. The *Optimistic Backtracking* method will try to locate the first incorrect parsing decision made by the parser, and replace this decision with the correct transition, and resume parsing from this state.

2 Related Work

Incremental deterministic classifier-based parsing algorithms have been studied in dependency parsing (Nivre and Scholz, 2004; Yamada and Matsumoto, 2003) and CFG parsing (Sagae and Lavie, 2005). Johansson and Nugues (2006) describe a non-deterministic implementation to the dependency parser outlined by Nivre and Scholz (2004), where they apply an *n*-best beam search strategy.

For a highly constrained unification-based formalism like HPSG, a deterministic parsing strategy could frequently lead to parse failures. Ninomiya et al. (2009) suggest an algorithm for deterministic shift-reduce parsing in HPSG. They outline two backtracking strategies for HPSG parsing. Their approach allows the parser to enter an old state if parsing fails or ends with non-sentential success, based on the minimal distance between the best candidate

Proceedings of the ACL-HLT 2011 Student Session, pages 58–63,
Portland, OR, USA 19-24 June 2011.

and the second best candidate in the sequence of transitions leading up to the current stage. Further constraints may be added, i.e. restricting the number of states the parser may backtrack. This algorithm is expanded by using a beam-thresholding best-first search algorithm, where each state in the parse has a state probability defined by the product of the probabilities of the selecting actions that has been taken to reach the state.

3 CuteForce

Optimistic Backtracking is in this paper used to evaluate CuteForce, an incremental deterministic HPSG parser currently in development. Similar to MaltParser (Nivre et al., 2007), it employs a classifier-based *oracle* to guide the shift-reduce parser that incrementally builds a syntactic/semantic HPSG derivation defined by LinGO English Resource Grammar (ERG) (Flickinger, 2000).

Parser Layout CuteForce has a more complex transition system than MaltParser in order to facilitate HPSG parsing. The sentence input buffer β is a list of tuples with token, part-of-speech tags and HPSG lexical types (i.e. supertags (Bangalore and Joshi, 1999)).

Given a set of ERG rules R and a sentence buffer β, a parser configuration is a tuple $c = (\alpha, \beta, \iota, \pi)$ where:

- α is a stack of "active" edges[1]
- β is a list of tuples of word forms W, part of speech tags POS and lexical types LT derived from a sentence $x = ((W_1, POS_1, LT_1), ...(W_n, POS_n, LT_n))$.
- ι is the current input position in β
- π is a stack of passive edges instantiating a ERG rule

The stack of passive edges π makes up the full HPSG representation of the input string if the string is accepted.

[1]An "active" edges in our sense is a hypothesis of an application of a binary rule where the left daughter is known (an element of π), and the specific binary ERG rule and the right daughter is yet to be found.

Transition System The shift-reduce parser has four different transitions, two of which are parameterized with a unary or binary ERG rule, which are added to the passive edges, hence building the HPSG structure. The four transitions are:

- ACTIVE – (adds an active edge to stack α, and increments ι)
- UNIT(R^1) – (adds unary passive edge to π instantiating unary ERG rule (R^1))
- PASSIVE(R^2) – (pops α and adds binary passive edge to π instantiating binary ERG rule (R^2))
- ACCEPT – (terminates the parse of the sentence. π represents the HPSG derivation of the sentence)

Derivation Example Figure 1 is a derivation example from Redwoods Treebank (Oepen et al., 2002). We note that the tree derivation consists of unary and binay productions, corresponding to the UNIT(R^1) and PASSIVE(R^2) parser transitions. Further, the pre-terminal lexical types have a *_le* suffix, and are provided together with the terminal word form in the input buffer for the parser.

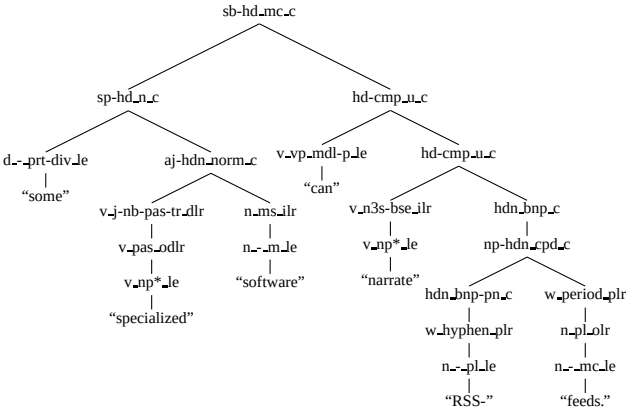

Figure 1: HPSG derivation from Redwoods Treebank.

Parsing Configuration Mode CuteForce can operate in three different oracle configurations: HPSG Unification mode, CFG approximation mode and unrestricted mode.

In HPSG Unification mode, the parser validates that no oracle decisions lead to an invalid HPSG derivation. All UNIT and PASSIVE transitions are

an implicit unification. For each parsing stage, the parsing oracle returns a ranked list of transitions. The highest-ranked transition not violating a unification constraint will be executed. If no transition yields a valid unification, parsing fails for the given sentence.

In CFG mode, a naive CFG approximation of the ERG is employed to guide the oracle. The CFG approximation consists of CFG rules harvested from the treebanks used in training the parser – for this purpose we have used existing Redwoods treebanks used in training, and augmented with derivations from WikiWoods, in total 300,000 sentences. Each ERG rule instantiation, using the identifiers shown in Figure 1 as non-terminal symbols, will be treated as a CFG rule, and each parser action will be validated against the set of CFG rules. If the parser action yields a CFG projection not found among the valid CFG rules in the CFG approximation, the CFG filter will block this transition. If the parser arrives at a state where the CFG filter blocks all further transitions, parsing fails.

In unrestricted mode, the oracle chooses the highest scoring transition without any further restrictions imposed. In this setting, the parser typically reaches close to 100 % coverage – the only sentences not covered in this setting are instances where the parser enters an infinite unit production loop. Hence, we will only evaluate the parser in CFG and Unification mode in this paper.

4 Optimistic Backtracking

Optimistic Backtracking can be added as an overlay to a transition-based parser in order to evaluate the parser in non-deterministic mode. The overlay has a linear time-complexity. This backtracking method is, to the best of our knowledge, the only method that applies ranking rather than some probability-based algorithm for backtracking. This aspect is critical for classification-based parsing oracles that do not yield a probability score in the ranking of candidate transitions.

Treating backtracking as a ranking problem has several attractive features. It may combine global and local syntactic and semantic information related to each candidate transition, contrary to a probabilistic approach that only employs the local transition probability. Utilizing global information also seems more sound from a human point of view. Consider sentence (1), it's first when the second verb (*fell*) is encountered that we would re-evaluate our original assumption, namely that *raced* may not be the head verb of the sentence. That *fell* indeed is a verb is surely relevant information for reconsidering *raced* as the head of a relative clause.

When the parser halts, the backtracker will rank each transition produced up until the point of failure according to which transition is most likely to be the first incorrect transition. When the best scoring transition is located, the parser will backtrack to this position, and replace this transition with the parsing oracle's second-best scoring transition for this current parsing state. If the parser later comes to another halt, only the transitions occurring after the first backtrack will be subject to change. Hence, the backtracker will always assume that its last backtrack was correct (thus being *Optimistic*). Having allowed the parser to backtrack unrestrictedly, we could theoretically have reached close to 100 % coverage, but the insights of parsing incrementally would have become less pronounced.

The search space for the backtracker is $n * m$ where n is the number of candidate transitions, and m is the total number of parser transitions. In *Optimistic Backtracking* we disregard the m dimension altogether by always choosing the second-best transition candidate ranked by the parsing oracle, assuming that the second-ranked transition in the given state actually was the correct transition. Hence we reduce the search-space to the n-dimension. In this paper, using CuteForce as HPSG parser, this assumption holds in about 80-90 % of the backtracks in CFG mode, in HPSG Unification mode this number is somewhat lower.

4.1 Baseline

As a baseline for identifying the incorrect transition, we use a strategy inspired by Ninomiya et al. (2009), namely to pick the candidate transition with the minimal probability difference between the best and the second best transition candidate. However, since we do not have true probability, a pseudo-probability is computed by taking the dot product of the feature vector and weight-vector for each best-scoring (P) and second-best scoring (P2) candidate transi-

tion, and use the proportion of the second-best score over the joint probability of the best and second-best scoring transition: $\frac{P2}{P+P2}$

In our development test set of 1794 sentences, we ran the parser in CFG and HPSG unification mode in deterministic and non-deterministic mode. The baseline results are found in Table 1 (CFG-BL) and Table 2 (UNI-BL). In CFG mode (Table 1), we obtain a 51.2 % reduction in parsing failure. In unification mode (Table 2) the parser is much more likely to fail, as the parse derivations are guaranteed to be a valid HPSG derivation. Baseline backtracking yields a mere 10 % reduction in parsing failures.

4.2 Feature Model

Each candidate transition is mapped to a feature vector that provides information about the transition. The task for the ranker is to identify the first incorrect transition in the sequence of transitions. The feature model used by the ranker employs features that can roughly be divided in three. First, the transition-specific features provide information on the nature of the candidate transition and surrounding transitions. Here we also have features related to the pseudo-probability of the transition (provided by the parsing oracle), and the oracle score distance between the best-scoring and second-best scoring transition for each given state. Secondly we have features related to the last token that was processed by the parser before it reached an invalid state, and the information on the incomplete HPSG derivation that was built at that state. These features are used in combination with local transition-specific features. Third, we have features concerning the preliminary HPSG derivation in the actual state of the transition.

Feature Types The list of transitions T = t_0, t_1, ... t_n comprises the candidate transitions that are subject to backtracking upon parsing failure. The feature types used by the backtracker includes:

- the pseudo-probability of the best scoring (P) and second best scoring (P2) transition
- the transition category of the current transition
- the probability proportion of the second best scoring transition over the joint probability $\frac{P2}{P+P2}$
- the transition number in the list of applicable candidates, and the number of remaining transitions, relative to the list of candidates
- the last lexical tag and part-of-speech tag that were processed before parsing failure
- the head category of the HPSG derivation and the left daughter unification candidate for the HPSG derivation in the current position
- the lexical tag relative to the current position in the buffer

The backtracker is trained as a linear SVM using SVM^{rank} (Joachims, 2006). Totally, the feature vector maps 24 features for each transition, including several combinations of the feature types above.

5 Evaluation

In this paper we trained CuteForce with data from Redwoods Treebank, augmented with derivations from WikiWoods (Flickinger et al., 2010). The test set contains a random sample of 1794 sentences from the Redwoods Treebank (which was excluded from the training data), with an average length of 14 tokens. Training data for the backtracker is extracted by parsing derivations from WikiWoods deterministically, and record transition candidates each time parsing fails, labeling the correct backtracking candidate, backtrack to this point, and resume parsing from this state.

5.1 Results

The first column (CFG-NB and UNI-NB) in Table 1 and 2 indicates the scores when the parser is run in deterministic mode, i.e. without backtracking. The second and third column contain results for baseline and *Optimistic* backtracking. *Coverage* refers to the proportion of sentences that received a parse. *Precision* refers to the backtracker's precision with respect to identifying the incorrect transition among the candidate transitions. $\sim$ *BT Cand* is the average number of candidate transitions the backtracker ranks when trying to predict the incorrect transition, and $\sim$ *BT Cand,1st* is the number of candidates at the initial point-of-failure. *Exact Matches* is the total number of parse derivations which are identical to the gold standard.

For *Ms per Sent* (milliseconds per sentence) it should be said that the code is not optimized, es-

pecially with respect to the HPSG unification algorithm[2]. How the figures relate to one another should however give a good indication on how the computational costs vary between the different configurations.

	CFG -NB	CFG -BL	CFG -Opt
Coverage	0.754	0.880	0.899
Precision	N/A	0.175	0.235
∼BT Cand	N/A	26.1	30.6
∼BT Cand,1st	N/A	51.5	51.5
Exact Matches	727	746	742
Ms per Sent	10.7	45.0	72.5

Table 1: Results – CFG mode

	UNI -NB	UNI -BL	UNI -Opt
Coverage	0.574	0.598	0.589
Precision	N/A	0.183	0.206
∼BT Cand	N/A	12.89	20.12
∼BT Cand,1st	N/A	51.6	51.6
Exact Matches	776	777	776
Ms per Sent	1801.4	5519.1	5345.2

Table 2: Results – HPSG unification mode

5.2 CFG approximation

The number of failed sentences is greatly reduced when backtracking is enabled. Using baseline backtracking, the reduction is 51.2 %, whereas *Optimistic* backtracking has a 59.1 % reduction in parse failures. Further, *Optimistic Backtracker* performs substantially better than baseline in identifying incorrect transitions.

The average number of candidate transitions ranged from 26 to 30 for the baseline and *Optimistic* backtracking strategy. It's interesting to observe that even with a success rate of about 1/5 in identifying the incorrect transition, the coverage is still greatly increased. That backtracking manages to recover so many sentences that initially failed, even if it does not manage to identify the incorrect transition, would seem to indicate that even when mistaken, the backtracker is producing a good prediction. On the other hand, the exact match score does not improve the same way as the coverage, this is directly related to the fact that the backtracker still has relatively low precision, as only a perfect prediction would leave the parser capable of deriving an exact match.

The success rate of about 0.23 in picking the incorrect transition in a set of in average 30 candidates indicates that treating the backtracking as a ranking problem is promising. The precision rate in itself is however relatively low, which serves as an indication of the difficulty of this task.

5.3 HPSG Unification

In unification mode the we see no substantive difference between deterministic mode, and baseline and *Optimistic* backtracking, and practically no improvement in the quality of the parses produced. In Table 2 we see that the only striking difference between the figures for the parser in backtracking mode and deterministic mode is the efficiency – the time consumption is increased by approximately a factor of 3.

5.4 Conclusion

The findings in this paper are specific to CuteForce. It is however very likely that the results would be similar for other deterministic HPSG parsers.

In CFG mode, the number of failed parses are more than halved compared to deterministic mode. It is likely that further increase could be obtained by relaxing constraints in the *Optimistic* algorithm.

In Unification mode, we experienced only a slight increase in coverage. By relaxing the *Optimistic* constraints, the time-complexity would go up. Considering how little the parser benefited from backtracking in unification mode with *Optimistic* constraints, it seems implausible that the parser will improve considerably without a heavy relaxation of the constraints in the *Optimistic* algorithm. If doing so, the attractive features of the parser's inherently deterministic nature will be overshadowed by a very large number of backtracks at a heavy computational cost. Hence, it's hard to see that such a semi-deterministic approach could have any advantages over other non-deterministic HPSG parsers neither in computational cost, performance or on a cognitive level.

[2]Specifically, the current unification back-end preforms non-destructive unification, i.e. it does not take advantage of the deterministic nature of CuteForce

Acknowledgements

The author would like to thank Stephan Oepen (University of Oslo) and Joakim Nivre (Uppsala University) for their valued input and inspiring feedback during the writing of this paper, and in the PhD project. Experimentation and engineering was made possible through access to the TITAN high-performance computing facilities at the University of Oslo (UiO), and we are grateful to the Scientific Computation staff at UiO, as well as to the Norwegian Metacenter for Computational Science.

References

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: an approach to almost parsing. *Computational Linguistics*, pages 237–265.

Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. 2010. Wikiwoods: Syntacto-semantic annotation for english wikipedia. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1):15 – 28.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM.

Richard Johansson and Pierre Nugues. 2006. Investigating multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 206–210. Association for Computational Linguistics.

Takashi Ninomiya, Nobuyuki Shimizu, Takuya Matsuzaki, and Hiroshi Nakagawa. 2009. Deterministic shift-reduce parsing for unification-based grammars by using default unification. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 603–611. Association for Computational Linguistics.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Chris Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods treebank. Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics*.

Kenji Sagae and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 125–132. Association for Computational Linguistics.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 195–206.

An Error Analysis of Relation Extraction in Social Media Documents

Gregory Ichneumon Brown
University of Colorado at Boulder
Boulder, Colorado
browngp@colorado.edu

Abstract

Relation extraction in documents allows the detection of how entities being discussed in a document are related to one another (e.g. part-of). This paper presents an analysis of a relation extraction system based on prior work but applied to the J.D. Power and Associates Sentiment Corpus to examine how the system works on documents from a range of social media. The results are examined on three different subsets of the JDPA Corpus, showing that the system performs much worse on documents from certain sources. The proposed explanation is that the features used are more appropriate to text with strong editorial standards than the informal writing style of blogs.

1 Introduction

To summarize accurately, determine the sentiment, or answer questions about a document it is often necessary to be able to determine the relationships between entities being discussed in the document (such as part-of or member-of). In the simple sentiment example

Example 1.1: I bought a new car yesterday. I love the powerful engine.

determining the sentiment the author is expressing about the car requires knowing that the engine is a part of the car so that the positive sentiment being expressed about the engine can also be attributed to the car.

In this paper we examine our preliminary results from applying a relation extraction system to the J.D. Power and Associates (JDPA) Sentiment Corpus (Kessler et al., 2010). Our system uses lexical features from prior work to classify relations, and we examine how the system works on different subsets from the JDPA Sentiment Corpus, breaking the source documents down into professionally written reviews, blog reviews, and social networking reviews. These three document types represent quite different writing styles, and we see significant difference in how the relation extraction system performs on the documents from different sources.

2 Relation Corpora

2.1 ACE-2004 Corpus

The Automatic Content Extraction (ACE) Corpus (Mitchell, et al., 2005) is one of the most common corpora for performing relation extraction. In addition to the co-reference annotations, the Corpus is annotated to indicate 23 different relations between real-world entities that are mentioned in the same sentence. The documents consist of broadcast news transcripts and newswire articles from a variety of news organizations.

2.2 JDPA Sentiment Corpus

The JDPA Corpus consists of 457 documents containing discussions about cars, and 180 documents discussing cameras (Kessler et al., 2010). In this work we only use the automotive documents. The documents are drawn from a variety of sources, and we particularly focus on the 24% of the documents from the JDPA Power Steering blog, 18% from Blogspot, and 18% from LiveJournal.

Proceedings of the ACL-HLT 2011 Student Session, pages 64–68,
Portland, OR, USA 19-24 June 2011.

The annotated mentions in the Corpus are single or multi-word expressions which refer to a particular real world or abstract entity. The mentions are annotated to indicate sets of mentions which constitute co-reference groups referring to the same entity. Five relationships are annotated between these entities: PartOf, FeatureOf, Produces, InstanceOf, and MemberOf. One significant difference between these relation annotations and those in the ACE Corpus is that the former are relations between sets of mentions (the co-reference groups) rather than between individual mentions. This means that these relations are not limited to being between mentions in the same sentence. So in Example 1.1, "engine" would be marked as a part of "car" in the JDPA Corpus annotations, but there would be no relation annotated in the ACE Corpus. For a more direct comparison to the ACE Corpus results, we restrict ourselves only to mentions within the same sentence (we discuss this decision further in section 5.4).

3 Relation Extraction System

3.1 Overview

The system extracts all pairs of mentions in a sentence, and then classifies each pair of mentions as either having a relationship, having an inverse relationship, or having no relationship. So for the PartOf relation in the JDPA Sentiment Corpus we consider both the relation "X is part of Y" and "Y is part of X". The classification of each mention pair is performed using a support vector machine implemented using libLinear (Fan et al., 2008).

To generate the features for each of the mention pairs a proprietary JDPA Tokenizer is used for parsing the document and the Stanford Parser (Klein and Manning, 2003) is used to generate parse trees and part of speech tags for the sentences in the documents.

3.2 Features

We used Zhou et al.'s lexical features (Zhou et al., 2005) as the basis for the features of our system similar to what other researchers have done (Chan and Roth, 2010). Additional work has extended these features (Jiang and Zhai, 2007) or incorporated other data sources (e.g. WordNet), but in this paper we focus solely on the initial step of applying these same lexical features to the JDPA Corpus.

The Mention Level, Overlap, Base Phrase Chunking, Dependency Tree, and Parse Tree features are the same as Zhou et al. (except for using the Stanford Parser rather than the Collins Parser). The minor changes we have made are summarized below:

- **Word Features**: Identical, except rather than using a heuristic to determine the head word of the phrase it is chosen to be the noun (or any other word if there are no nouns in the mention) that is the least deep in the parse tree. This change has minimal impact.
- **Entity Types**: Some of the entity types in the JDPA Corpus indicate the type of the relation (e.g. CarFeature, CarPart) and so we replace those entity types with "Unknown".
- **Token Class**: We added an additional feature (TC12+ET12) indicating the Token Class of the head words (e.g. Abbreviation, DollarAmmount, Honorific) combined with the entity types.
- **Semantic Information**: These features are specific to the ACE relations and so are not used. In Zhou et al.'s work, this set of features increases the overall F-Measure by 1.5.

4 Results

4.1 ACE Corpus Results

We ran our system on the ACE-2004 Corpus as a baseline to prove that the system worked properly and could approximately duplicate Zhou et al.'s results. Using 5-fold cross validation on the newswire and broadcast news documents in the dataset we achieved an average overall F-Measure of 50.6 on the fine-grained relations. Although a bit lower than Zhou et al.'s result of 55.5 (Zhou et al., 2005), we attribute the difference to our use of a different tokenizer, different parser, and having not used the semantic information features.

4.2 JDPA Sentiment Corpus Results

We randomly divided the JDPA Corpus into training (70%), development (10%), and test (20%) datasets. Table 1 shows relation extraction results of the system on the test portion of the corpus. The results are further broken out by three different source types to highlight the differences caused

Relation	All Documents			LiveJournal			Blogspot			JDPA		
	P	**R**	**F**	P	R	F	P	R	F	P	R	F
FEATURE OF	44.8	42.3	43.5	26.8	35.8	30.6	44.1	40.0	42.0	59.0	55.0	56.9
MEMBER OF	34.1	10.7	16.3	0.0	0.0	0.0	36.0	13.2	19.4	36.4	13.7	19.9
PART OF	46.5	34.7	39.8	41.4	17.5	24.6	48.1	35.6	40.9	48.8	43.9	46.2
PRODUCES	51.7	49.2	50.4	05.0	36.4	08.8	43.7	36.0	39.5	66.5	64.6	65.6
INSTANCE OF	37.1	16.7	23.0	44.8	14.9	22.4	42.1	13.0	19.9	30.9	29.6	30.2
Overall	46.0	36.2	**40.5**	27.1	22.6	**24.6**	45.2	33.3	**38.3**	53.7	46.5	**49.9**

Table 1: Relation extraction results on the JDPA Corpus test set, broken down by document source.

	LiveJournal	Blogspot	JDPA	ACE
Tokens Per Sentence	**19.2**	18.6	16.5	19.7
Relations Per Sentence	1.08	1.71	**2.56**	0.56
Relations Not In Same Sentence	33%	30%	**27%**	0%
Training Mention Pairs in One Sentence	58,452	54,480	**95,630**	77,572
Mentions Per Sentence	**4.26**	**4.32**	4.03	3.16
Mentions Per Entity	**1.73**	1.63	1.33	2.36
Mentions With Only One Token	**77.3%**	73.2%	61.2%	56.2%

Table 2: Selected document statistics for three JDPA Corpus document sources.

by the writing styles from different types of media: LiveJournal (livejournal.com), a social media site where users comment and discuss stories with each other; Blogspot (blospot.com), Google's blogging platform; and JDPA (jdpower.com's Power Steering blog), consisting of reviews of cars written by JDPA professional writers/analysts. These subsets were selected because they provide the extreme (JDPA and LiveJournal) and average (Blogspot) results for the overall dataset.

5 Analysis

Overall the system is not performing as well as it does on the ACE-2004 dataset. However, there is a 25 point F-Measure difference between the LiveJournal and JDPA authored documents. This suggests that the informal style of the LiveJournal documents may be reducing the effectiveness of the features developed by Zhou et al., which were developed on newswire and broadcast news transcript documents.

In the remainder of this section we look at a statistical analysis of the training portion of the JDPA Corpus, separated by document source, and suggest areas where improved features may be able to aid relation extraction on the JDPA Corpus.

5.1 Document Statistic Effects on Classifier

Table 2 summarizes some important statistical differences between the documents from different sources. These differences suggest two reasons why the instances being used to train the classifier could be skewed disproportionately towards the JDPA authored documents.

First, the JDPA written documents express a much larger number of relations between entities. When training the classifier, these differences will cause a large share of the instances that have a relation to be from a JDPA written document, skewing the classifier towards any language clues specific to these documents.

Second, the number of mention pairs occurring within one sentence is significantly higher in the JDPA authored documents than the other documents. This disparity is even true on a per sentence or per document basis. This provides the classifier with significantly more negative examples written in a JDPA written style.

LiveJournal		Blogspot		JDPA	
Mention Phrase	%	Mention Phrase	%	Mention Phrase	%
car	6.2	it	8.1	features	2.4
Maybach	5.6	car	2.1	vehicles	1.6
it	3.7	its	2.0	its	1.4
it's	1.7	cars	2.0	Journey	1.3
Maybach 57 S	1.5	Hyundai	2.0	car	1.2
It	1.2	vehicle	1.5	2 T Sport	1.2
mileage	1.1	one	1.5	G37	1.2
its	1.1	engine	1.5	models	1.1
engine	0.9	power	1.1	engine	1.1
57 S	0.9	interior	1.1	It	1.1
Total: 23.9%		**Total: 22.9%**		**Total: 13.6%**	

Table 3: Top 10 phrases in mention pairs whose relation was incorrectly classified, and the total percentage of errors from the top ten.

Word	Percent of All Tokens in Documents			
	LiveJournal	Blogspot	JDPA	ACE
car	**0.86**	0.71	0.20	0.01
I	**1.91**	1.28	0.24	0.21
it	**1.42**	0.97	0.23	0.63
It	0.33	0.27	**0.35**	0.09
its	**0.25**	0.18	0.22	0.19
the	**4.43**	**4.60**	3.54	4.81

Table 4: Frequency of some common words per token.

POS	POS Occurrence Per Sentence			
	LiveJournal	Blogspot	JDPA	ACE
NN	2.68	3.01	**3.21**	2.90
NNS	0.68	0.73	**0.85**	1.08
NNP	0.93	1.41	**1.89**	1.48
NNPS	0.03	0.03	0.03	0.06
PRP	**0.98**	0.70	0.20	0.57
PRP$	**0.21**	0.18	0.07	0.20

Table 5: Frequency of select part-of-speech tags.

5.2 Common Errors

Table 3 shows the mention phrases that occur most commonly in the incorrectly classified mention pairs. For the LiveJournal and Blogspot data, many more of the errors are due to a few specific phrases being classified incorrectly such as "car", "Maybach", and various forms of "it". The top four phrases constitute 17% of the errors for LiveJournal and 14% for Blogspot. Whereas the JDPA documents have the errors spread more evenly across mention phrases, with the top 10 phrases constituting 13.6% of the total errors.

Furthermore, the phrases causing many of the problems for the LiveJournal and Blogspot relation detection are generic nouns and pronouns such as "car" and "it". This suggests that the classifier is having difficulty determining relationships when these less descriptive words are involved.

5.3 Vocabulary

To investigate where these variations in phrase error rates comes from, we performed two analyses of the word frequencies in the documents: Table 4 shows the frequency of some common words in the documents; Table 5 shows the frequency of a select set of parts-of-speech per sentence in the document.

We find that despite all the documents discussing cars, the JDPA reviews use the word "car" much less often, and use proper nouns significantly more often. Although "car" also appears in the top ten errors on the JDPA documents, the total percentage of the errors is one fifth of the error rate on the LiveJournal documents. The JDPA authored documents also tend to have more multi-word mention phrases (Table 2) suggesting that the authors use more descriptive language when referring to an entity. 77.3% of the mentions in LiveJournal documents use only a single word while 61.2% of mentions JDPA authored documents are a single word.

Rather than descriptive noun phrases, the LiveJournal and Blogspot documents make more use of pronouns. LiveJournal especially uses pronouns often, to the point of averaging one per sentence, while JDPA uses only one every five sentences.

5.4 Extra-Sentential Relations

Many relations in the JDPA Corpus occur between entities which are not mentioned in the same sentence. Our system only detects relations between mentions in the same sentence, causing about 29% of entity relations to never be detected (Table 2).

The LiveJournal documents are more likely to contain relationships between entities that are not mentioned in the same sentence. In the semantic role labeling (SRL) domain, extra-sentential arguments have been shown to significantly improve SRL performance (Gerber and Chai, 2010). Improvements in entity relation extraction could likely be made by extending Zhou et al.'s features across sentences.

6 Conclusion

The above analysis shows that at least some of the reason for the system performing worse on the JDPA Corpus than on the ACE-2004 Corpus is that many of the documents in the JDPA Corpus have a different writing style from the news articles in the ACE Corpus. Both the ACE news documents, and the JDPA authored documents are written by professional writers with stronger editorial standards than the other JDPA Corpus documents, and the relation extraction system performs much better on professionally edited documents. The heavy use of pronouns and less descriptive mention phrases in the other documents seems to be one cause of the reduction in relation extraction performance. There is also some evidence that because of the greater number of relations in the JPDA authored documents that the classifier training data could be skewed more towards those documents.

Future work needs to explore features that can address the difference in language usage that the different authors use. This work also does not address whether the relation extraction task is being negatively impacted by poor tokenization or parsing of the documents rather than the problems being caused by the relation classification itself. Further work is also needed to classify extra-sentential relations, as the current methods look only at relations occurring within a single sentence thus ignoring a large percentage of relations between entities.

Acknowledgments

This work was partially funded and supported by J. D. Power and Associates. I would like to thank Nicholas Nicolov, Jason Kessler, and Will Headden for their help in formulating this work, and my thesis advisers: Jim Martin, Rodney Nielsen, and Mike Mozer.

References

Chan, Y. S. and Roth D. *Exploiting Background Knowledge for Relation Extraction*. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010).

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. *LIBLINEAR: A library for large linear classification*. Journal of Machine Learning Research 9(2008), 1871-1874. 2008.

Gerber, M. and Chai, J. *Beyond NomBank: A Study of Implicit Arguments for Nominal Predicates*. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1583-1592. 2010.

Jiang, J. and Zhai, C.X. *A systematic exploration of the feature space for relation extraction*. In The Proceedings of NAACL/HLT. 2007.

Kessler J., Eckert M., Clark L., and Nicolov N.. *The ICWSM 2010 JDPA Sentiment Corpus for the Automotive Domain* International AAAI Conference on Weblogs and Social Media Data Challenge Workshop. 2010.

Klein D. and Manning C. *Accurate Unlexicalized Parsing*. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430. 2003.

Mitchell A., et al. *ACE 2004 Multilingual Training Corpus*. Linguistic Data Consortium, Philadelphia. 2005.

Zhou G., Su J., Zhang J., and Zhang M. *Exploring various knowledge in relation extraction*. Proceedings of the 43rd Annual Meeting of the ACL. 2005.

Effects of Noun Phrase Bracketing in Dependency Parsing and Machine Translation

Nathan Green
Charles University in Prague
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
green@ufal.mff.cuni.cz

Abstract

Flat noun phrase structure was, up until recently, the standard in annotation for the Penn Treebanks. With the recent addition of internal noun phrase annotation, dependency parsing and applications down the NLP pipeline are likely affected. Some machine translation systems, such as TectoMT, use deep syntax as a language transfer layer. It is proposed that changes to the noun phrase dependency parse will have a cascading effect down the NLP pipeline and in the end, improve machine translation output, even with a reduction in parser accuracy that the noun phrase structure might cause. This paper examines this noun phrase structure's effect on dependency parsing, in English, with a maximum spanning tree parser and shows a 2.43%, 0.23 Bleu score, improvement for English to Czech machine translation.

1 Introduction

Noun phrase structure in the Penn Treebank has up until recently been only considered, due to underspecification, a flat structure. Due to the annotation and work of Vadas and Curran (2007a; 2007b; 2008), we are now able to create Natural Language Processing (NLP) systems that take advantage of the internal structure of noun phrases in the Penn Treebank. This extra internal structure introduces additional complications in NLP applications such as parsing.

Dependency parsing has been a prime focus of NLP research of late due to its ability to help parse languages with a free word order. Dependency parsing has been shown to improve NLP systems in certain languages and in many cases is considered the state of the art in the field. Dependency parsing made many improvements due to the CoNLL X shared task (Buchholz and Marsi, 2006). However, in most cases, these systems were trained with a flat noun phrase structure in the Penn Treebank. Vadas' internal noun phrase structure has been used in previous work on constituent parsing using Collin's parser (Vadas and Curran, 2007c), but has yet to be analyzed for its effects on dependency parsing.

Parsing is very early in the NLP pipeline. Therefore, improvements in parsing output could have an improvement on other areas of NLP in many cases, such as Machine Translation. At the same time, any errors in parsing will tend to propagate down the NLP pipeline. One would expect parsing accuracy to be reduced when the complexity of the parse is increased, such as adding noun phrase structure. But, for a machine translation system that is reliant on parsing, the new noun phrase structure, even with reduced parser accuracy, may yield improvements due to a more detailed grammatical structure. This is particularly of interest for dependency relations, as it may aid in finding the correct head of a term in a complex noun phrase.

This paper examines the results and errors in parsing and machine translation of dependency parsers, trained with annotated noun phrase structure, against those with a flat noun phrase structure. These results are compared with two systems: a Baseline Parser with no internally annotated noun phrases and a Gold NP Parser trained with data which contains

Proceedings of the ACL-HLT 2011 Student Session, pages 69–74,
Portland, OR, USA 19-24 June 2011.

gold standard internal noun phrase structure annotation. Additionally, we analyze the effect of these improvements and errors in parsing down the NLP pipeline on the TectoMT machine translation system (Žabokrtský et al., 2008).

Section 2 contains background information needed to understand the individual components of the experiments. The methodology used to carry out the experiments is described in Section 3. Results are shown and discussed in Section 4. Section 5 concludes and discusses future work and implications of this research.

2 Related Work

2.1 Dependency Parsing

Dependence parsing is an alternative view to the common phrase or constituent parsing techniques used with the Penn Treebank. Dependency relations can be used in many applications and have been shown to be quite useful in languages with a free word order. With the influx of many data-driven techniques, the need for annotated dependency relations is apparent. Since there are many data sets with constituent relations annotated, this paper uses free conversion software provided from the CoNLL 2008 shared task to create dependency relations (Johansson and Nugues, 2007; Surdeanu et al., 2008).

2.2 Dependency Parsers

Dependency parsing comes in two main forms: Graph algorithms and Greedy algorithms. The two most popular algorithms are McDonald's MSTParser (McDonald et al., 2005) and Nivre's MaltParser (Nivre, 2003). Each parser has its advantages and disadvantages, but the accuracy overall is approximately the same. The types of errors made by each parser, however, are very different. MSTParser is globally trained for an optimal solution and this has led it to get the best results on longer sentences. MaltParser on the other hand, is a greedy algorithm. This allows it to perform extremely well on shorter sentences, as the errors tend to propagate and cause more egregious errors in longer sentences with longer dependencies (McDonald and Nivre, 2007). We expect each parser to have different errors handling internal noun phrase structure, but for this paper we will only be examining the globally trained MSTParser.

2.3 TectoMT

TectoMT is a machine translation framework based on Praguian tectogrammatics (Sgall, 1967) which represents four main layers: word layer, morphological layer, analytical layer, and tectogrammatical layer (Popel et al., 2010). This framework is primarily focused on the translation from English into Czech. Since much of dependency parsing work has been focused on Czech, this choice of machine translation framework logically follows as TectoMT makes direct use of the dependency relationships. The work in this paper primarily addresses the noun phrase structure in the analytical layer (SEnglishA in Figure 1).

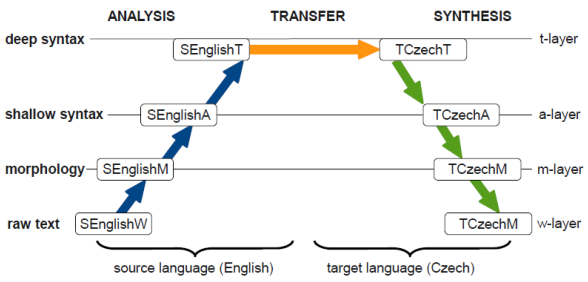

Figure 1: Translation Process in TectoMT in which the tectogrammatical layer is transfered from English to Czech.

TectoMT is a modular framework built in Perl. This allows great ease in adding the two different parsers into the framework since each experiment can be run as a separate "Scenario" comprised of different parsing "Blocks". This allows a simple comparison of two machine translation system in which everything remains constant except the dependency parser.

2.4 Noun Phrase Structure

The Penn Treebank is one of the most well known English language treebanks (Marcus et al., 1993), consisting of annotated portions of the Wall Street Journal. Much of the annotation task is painstakingly done by annotators in great detail. Some structures are not dealt with in detail, such as noun phrase structure. Not having this information makes it difficult to tell the dependencies on phrases such as

"crude oil prices" (Vadas and Curran, 2007c). Without internal annotation it is ambiguous whether the phrase is stating "crude prices" (crude (oil prices)) or "crude oil" ((crude oil) prices).

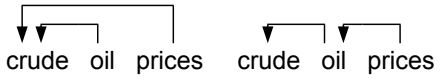

Figure 2: Ambiguous dependency caused by internal noun phrase structure.

Manual annotation of these phrases would be quite time consuming and as seen in the example above, sometimes ambiguous and therefore prone to poor inter-annotator agreement. Vadas and Curran have constructed a Gold standard version Penn treebank with these structures. They were also able to train supervised learners to an F-score of 91.44% (Vadas and Curran, 2007a; Vadas and Curran, 2007b; Vadas and Curran, 2008). The additional complexity of noun phrase structure has been shown to reduce parser accuracy in Collin's parser but no similar evaluation has been conducted for dependency parsers. The internal noun phrase structure has been used in experiments prior but without evaluation with respect to the noun phrases (Galley and Manning, 2009).

3 Methodology

The Noun Phrase Bracketing experiments consist of a comparison two systems.

1. The Baseline system is McDonald's MSTParser trained on the Penn Treebank in English without any extra noun phrase bracketing.

2. The Gold NP Parser is McDonald's MSTParser trained on the Penn Treebank in English with gold standard noun phrase structure annotations (Vadas and Curran, 2007a).

3.1 Data Sets

To maintain a consistent dataset to compare to previous work we use the Wall Street Journal (WSJ) section of the Penn Treebank since it was used in the CoNLL X shared task on dependency parsing (Buchholz and Marsi, 2006). Using the same common breakdown of datasets, we use WST section 02-21 for training and section 22 for testing, which allows us to have comparable results to previous works. To test the effects of the noun phrase structure on machine translation, ACL 2008's Workshop on Statistical Machine translation's (WMT) data are used.

3.2 Process Flow

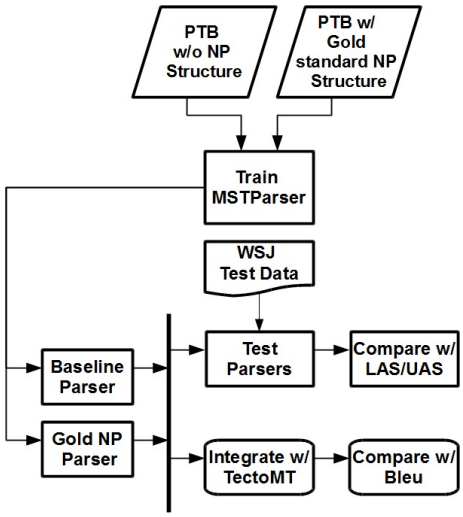

Figure 3: Experiment Process Flow. PTB (Penn Tree Bank), NP (Noun Phrase Structure), LAS (Labeled Accuracy Score), UAS (Unlabeled Accuracy Score), Wall Street Journal (WSJ)

We begin the the experiments by constructing two data sets:

1. The Penn Treebank with no internal noun phrase structure (PTB w/o NP structure).

2. The Penn Treebank with gold standard noun phrase annotations provided by Vadas and Curran (PTB w/ gold standard NP structure).

From these datasets we construct two separate parsers. These parsers are trained using McDonald's Maximum Spanning Tree Algorithm (MSTParser) (McDonald et al., 2005).

Both of the parsers are then tested on a subset of the WSJ corpus, section 22, of the Penn Treebank and the UAS and LAS scores are generated. Errors generated by each of these systems are then compared to discover where the internal noun phrase structure affects the output. Parser accuracy is not necessarily the most important aspect of this work.

The effect of this noun phrase structure down the NLP pipeline is also crucial. For this, the parsers are inserted into the TectoMT system.

3.3 Metrics

*L*abeled *A*ccuracy *S*core (LAS) and *U*nlabeled *A*ccuracy *S*core (UAS) are the primary ways to evaluate dependency parsers. UAS is the percentage of words that are correctly linked to their heads. LAS is the percentage of words that are connected to their correct heads and have the correct dependency label. UAS and LAS are used to compare one system against another, as was done in CoNLL X (Buchholz and Marsi, 2006).

The Bleu (*Bi*L*ingual* *E*valuation *U*nderstudy) score is an automatic scoring mechanism for machine translation that is quick and can be reused as a benchmark across machine translation tasks. Bleu is calculated as the geometric mean of n-grams comparing a machine translation and a reference text (Papineni et al., 2002). This experiment compares the two parsing systems against each other using the above metrics. In both cases the test set data is sampled 1,000 times without replacement to calculate statistical significance using a pairwise comparison.

4 Results and Discussion

When applied, the gold standard annotations changed approximately 1.5% of the edges in the training data. Once trained, both parsers were tested against section 22 of their respective annotated corpora. As Table 1 shows, the Baseline Parser obtained near identical LAS and UAS scores. This was expected given the additional complexity of predicting the noun phrase structure and the previous work on noun phrase bracketing's effect on Collin's parser.

Systems	LAS	UAS
Baseline Parser	88.12%	91.11%
Gold NP Parser	88.10%	91.10%

Table 1: Parsing results for the Baseline and Gold NP Parsers. Each is trained on Section 02-21 of the WSJ and tested on Section 22

While possibly more error prone, the 1.5% change in edges in the training data did appear to add more useful syntactic structure to the resulting parses as can be seen in Table 2. With the additional noun phrase bracketing, the resulting Bleu score increased 0.23 points or 2.43%. The improvement is statistically significant with 95% confidence using pairwise bootstrapping of 1,000 test sets randomly sampled with replacement (Koehn, 2004; Zhang et al., 2004). In Figure 4 we can see that the difference between each of the 1,000 samples was above 0, meaning the Gold NP Parser performed consistently better given each sample.

Systems	Bleu
Baseline Parser	9.47
Gold NP Parser	**9.70**

Table 2: TectoMT results of a complete system run with both the Baseline Parser and Gold NP Parser. Both are tested on WMT08 data. Results are an average of 1,000 bootstrapped test sets with replacement.

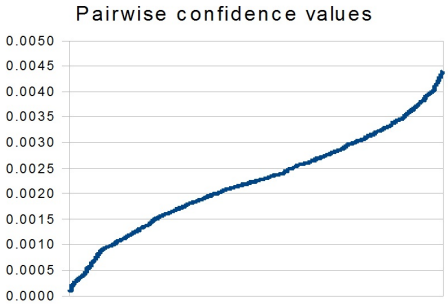

Figure 4: The Gold NP Parser shows statistically significant improvement with 95% confidence. The difference in Bleu score is represented on the Y-axis and the bootstrap iteration is displayed on the X-axis. The samples were sorted by the difference in bleu score.

Visually, changes can be seen in the English side parse that affect the overall translation quality. Sentences that contained incorrect noun phrase structure such as "The second vice-president and Economy minister, Pedro Solbes" as seen in Figure 5 and Figure 6 were more correctly parsed in the Gold NP Parser. In Figure 5 "and" is incorrectly assigned to the bottom of a noun phrase and does not connect any segments together in the output of the Baseline Parser, while it connects two phrases in Figure 6 which is the output of the Gold NP Parser. This shift in bracketing also allows the proper noun, which is shaded, to be assigned to the correct head, the rightmost noun in the phrase.

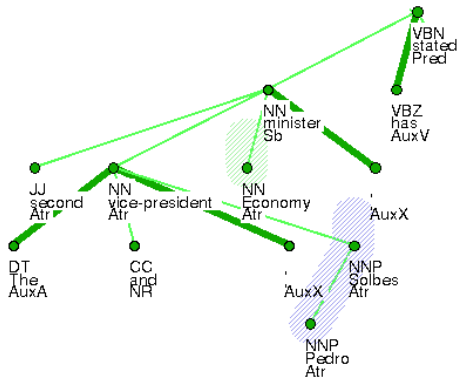

Figure 5: The parse created with the data with flat structures does not appear to handle noun phrases with more depth, in this case the 'and' does not properly connect the two components.

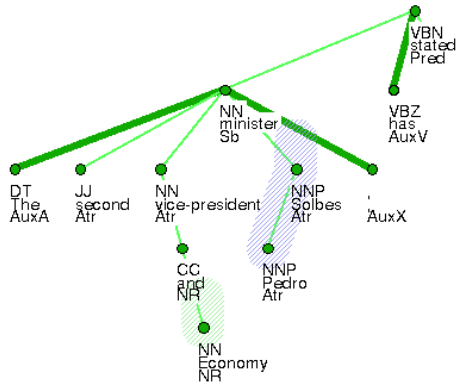

Figure 6: With the addition of noun phrase structure in parser, the complicated noun phrase appears to be better structured. The 'and' connects two components instead of improperly being a leaf node.

5 Conclusion

This paper has demonstrated the benefit of additional noun phrase bracketing in training data for use in dependency parsing and machine translation. Using the additional structure, the dependency parser's accuracy was minimally reduced. Despite this reduction, machine translation, much further down the NLP pipeline, obtained a 2.43% jump in Bleu score and is statistically significant with 95% confidence. Future work should examine similar experiments with MaltParser and other machine translation systems.

6 Acknowledgements

This research has received funding from the European Commissions 7th Framework Program (FP7) under grant agreement n° 238405 (CLARA), and from grant MSM 0021620838. I would like to thank Zdeněk Žabokrtský for his guidance in this research and also the anonymous reviewers for their comments.

References

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164, Morristown, NJ, USA. Association for Computational Linguistics.

Michel Galley and Christopher D. Manning. 2009. Quadratic-time dependency parsing for machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 773–781, Suntec, Singapore, August. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*, pages 105–112, Tartu, Estonia, May 25-26.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19:313–330, June.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530, Morristown, NJ, USA. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*, pages 149–160.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Martin Popel, Zdeněk Žabokrtský, and Jan Ptáček. 2010. Tectomt: Modular nlp framework. In *IceTAL*, pages 293–304.

Petr Sgall. 1967. *Generativní popis jazyka a česká deklinace*. Academia, Prague, Czech Republic.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pages 159–177, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Vadas and James Curran. 2007a. Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247, Prague, Czech Republic, June. Association for Computational Linguistics.

David Vadas and James R. Curran. 2007b. Large-scale supervised models for noun phrase bracketing. In *Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 104–112, Melbourne, Australia, September.

David Vadas and James R. Curran. 2007c. Parsing internal noun phrase structure with collins' models. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 109–116, Melbourne, Australia, December.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with CCG. In *Proceedings of ACL-08: HLT*, pages 335–343, Columbus, Ohio, June. Association for Computational Linguistics.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. Tectomt: highly modular mt system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 167–170, Morristown, NJ, USA. Association for Computational Linguistics.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system. In *In Proceedings of Proceedings of Language Resources and Evaluation (LREC-2004*, pages 2051–2054.

Towards a Framework for Abstractive Summarization of Multimodal Documents

Charles F. Greenbacker
Dept. of Computer & Information Sciences
University of Delaware
Newark, Delaware, USA
charlieg@cis.udel.edu

Abstract

We propose a framework for generating an abstractive summary from a semantic model of a multimodal document. We discuss the type of model required, the means by which it can be constructed, how the content of the model is rated and selected, and the method of realizing novel sentences for the summary. To this end, we introduce a metric called *information density* used for gauging the importance of content obtained from text and graphical sources.

1 Introduction

The automatic summarization of text is a prominent task in the field of natural language processing (NLP). While significant achievements have been made using statistical analysis and sentence extraction, "true abstractive summarization remains a researcher's dream" (Radev et al., 2002). Although existing systems produce high-quality summaries of relatively simple articles, there are limitations as to the types of documents these systems can handle.

One such limitation is the summarization of multimodal documents: no existing system is able to incorporate the non-text portions of a document (e.g., information graphics, images) into the overall summary. Carberry et al. (2006) showed that the content of information graphics is often not repeated in the article's text, meaning important information may be overlooked if the graphical content is not included in the summary. Systems that perform statistical analysis of text and extract sentences from the original article to assemble a summary cannot access the information contained in non-text components, let alone seamlessly combine that information with the extracted text. The problem is that information from the text and graphical components can only be integrated at the *conceptual* level, necessitating a semantic understanding of the underlying concepts.

Our proposed framework enables the generation of abstractive summaries from unified semantic models, regardless of the original format of the information sources. We contend that this framework is more akin to the human process of conceptual integration and regeneration in writing an abstract, as compared to the traditional NLP techniques of rating and extracting sentences to form a summary. Furthermore, this approach enables us to generate summary sentences about the information collected from graphical formats, for which there are no sentences available for extraction, and helps avoid the issues of coherence and ambiguity that tend to affect extraction-based summaries (Nenkova, 2006).

2 Related Work

Summarization is generally seen as a two-phase process: identifying the important elements of the document, and then using those elements to construct a summary. Most work in this area has focused on extractive summarization, assembling the summary from sentences representing the information in a document (Kupiec et al., 1995). Statistical methods are often employed to find key words and phrases (Witbrock and Mittal, 1999). Discourse structure (Marcu, 1997) also helps indicate the most important sentences. Various machine learning techniques have been applied (Aone et al., 1999; Lin, 1999), as well as approaches combining surface, content, rel-

Proceedings of the ACL-HLT 2011 Student Session, pages 75–80,
Portland, OR, USA 19-24 June 2011.

evance and event features (Wong et al., 2008).

However, a few efforts have been directed towards abstractive summaries, including the modification (i.e., editing and rewriting) of extracted sentences (Jing and McKeown, 1999) and the generation of novel sentences based on a deeper understanding of the concepts being described. Lexical chains, which capture relationships between related terms in a document, have shown promise as an intermediate representation for producing summaries (Barzilay and Elhadad, 1997). Our work shares similarities with the knowledge-based text condensation model of Reimer and Hahn (1988), as well as with Rau et al. (1989), who developed an information extraction approach for conceptual information summarization. While we also build a conceptual model, we believe our method of construction will produce a richer representation. Moreover, Reimer and Hahn did not actually produce a natural language summary, but rather a condensed text graph.

Efforts towards the summarization of multimodal documents have included naïve approaches relying on image captions and direct references to the image in the text (Bhatia et al., 2009), while content-based image analysis and NLP techniques are being combined for multimodal document indexing and retrieval in the medical domain (Névéol et al., 2009).

3 Method

Our method consists of the following steps: building the semantic model, rating the informational content, and generating a summary. We construct the semantic model in a knowledge representation based on typed, structured objects organized under a foundational ontology (McDonald, 2000). To analyze the text, we use Sparser,[1] a linguistically-sound, phrase structure-based chart parser with an extensive and extendible semantic grammar (McDonald, 1992). For the purposes of this proposal, we assume a relatively complete semantic grammar exists for the domain of documents to be summarized. In the prototype implementation (currently in progress), we are manually extending an existing grammar on an as-needed basis, with plans for large-scale learning of new rules and ontology definitions as future work. Projects like the Never-Ending Language Learner (Carlson et al., 2010) may enable us to induce these resources automatically.

Although our framework is general enough to cover any image type, as well as other modalities (e.g., audio, video), since image understanding research has not yet developed tools capable of extracting semantic content from every possible image, we must restrict our focus to a limited class of images for the prototype implementation. Information graphics, such as bar charts and line graphs, are commonly found in popular media (e.g., magazines, newspapers) accompanying article text. To integrate this graphical content, we use the SIGHT system (Demir et al., 2010b) which identifies the intended message of a bar chart or line graph along with other salient propositions conveyed by the graphic. Extending the prototype to incorporate other modalities would not entail a significant change to the framework. However, it would require adding a module capable of mapping the particular modality to its underlying message-level semantic content.

The next sections provide detail regarding the steps of our method, which will be illustrated on a short article from the May 29, 2006 edition of Businessweek magazine entitled, "Will Medtronic's Pulse Quicken?"[2] This particular article was chosen due to good coverage in the existing Sparser grammar for the business news domain, and because it appears in the corpus of multimodal documents made available by the SIGHT project.

3.1 Semantic Modeling

Figure 1 shows a high-level (low-detail) overview of the type of semantic model we can build using Sparser and SIGHT. This particular example models the article text (including title) and line graph from the Medtronic article. Each box represents an individual concept recognized in the document. Lines connecting boxes correspond to relationships between concepts. In the interest of space, the individual attributes of the model entries have been omitted from this diagram, but are available in Figure 2, which zooms into a fragment of the model showing the concepts that are eventually rated most salient (Section 3.2) and selected for inclusion in

[1] https://github.com/charlieg/Sparser

[2] Available at http://www.businessweek.com/magazine/content/06_22/b3986120.htm.

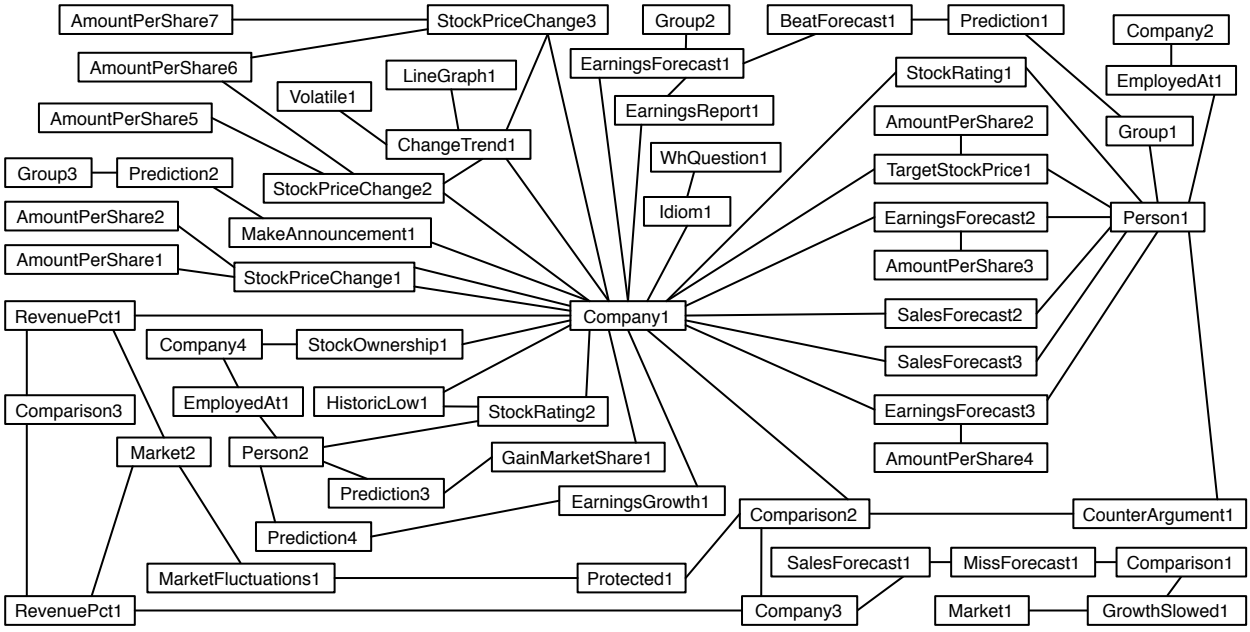

Figure 1: High-level overview of semantic model for Medtronic article.

the summary (Section 3.3). The top portion of each box in Figure 2 indicates the name of the conceptual category (with a number to distinguish between instances), the middle portion shows various attributes of the concept with their values, and the bottom portion contains some of the original phrasings from the text that were used to express these concepts (formally stored as a synchronous TAG) (McDonald and Greenbacker, 2010)). Attribute values in angle brackets (<>) are references to other concepts, hash symbols (#) refer to a concept or category that has not been instantiated in the current model, and each expression is preceded by a sentence tag (e.g., "P1S4" stands for "paragraph 1, sentence 4").

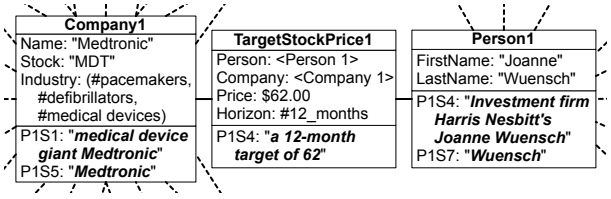

Figure 2: Detail of Figure 1 showing concepts rated most important and selected for inclusion in the summary.

As illustrated in this example, concepts conveyed by the graphics in the document can also be included in the semantic model. The overall intended message (ChangeTrend1) and additional propositions (Volatile1, StockPriceChange3, etc.) that SIGHT extracts from the line graph and deems important are added to the model produced by Sparser by simply inserting new concepts, filling slots for existing concepts, and creating new connections. This way, information gathered from both text and graphical sources can be integrated at the *conceptual* level regardless of the format of the source.

3.2 Rating Content

Once document analysis is complete and the semantic model has been built, we must determine which concepts conveyed by the document and captured in the model are most salient. Intuitively, the concepts containing the most information and having the most connections to other important concepts in the model are those we'd like to convey in the summary. We propose the use of an *information density* metric (ID) which rates a concept's importance based on a number of factors:[3]

- Completeness of attributes: the concept's filled-in slots (f) vs. its total slots (s) ["saturation level"], and the importance of the concepts (c_i) filling these slots [a recursive value]:

$$\frac{f}{s} * log(s) * \sum_{i=1}^{f} ID(c_i)$$

[3]The first three factors are similar to the dominant slot fillers, connectivity patterns, and frequency criteria described by Reimer and Hahn (1988).

- Number of connections/relationships (n) with other concepts (c_j), and the importance of these connected concepts [a recursive value]:

$$\sum_{j=1}^{n} ID(c_j)$$

- Number of expressions (e) realizing the concept in the current document

- Prominence based on document and rhetorical structure (W_D & W_R), and salience assessed by the graph understanding system (W_G)

Saturation refers to the level of completeness with which the knowledge base entry for a given concept is "filled-out" by information obtained from the document. As information is collected about a concept, the corresponding slots in its concept model entry are assigned values. The more slots that are filled, the more we know about a given instance of a concept. When all slots are filled, the model entry for that concept is "complete," at least as far as the ontological definition of the concept category is concerned. As saturation level is sensitive to the amount of detail in the ontology definition, this factor must be normalized by the number of attribute slots in its definition, thus $log(s)$ above.

In Figure 3 we can see an example of relative saturation level by comparing the attribute slots for Company2 with that of Company1 in Figure 2. Since the "Stock" slot is filled for Medtronic and remains empty for Harris Nesbitt, we say that the concept for Company1 is more saturated (i.e., more complete) than that of Company2.

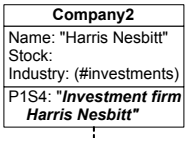

Figure 3: Detail of Figure 1 showing example concept with unfilled attribute slot.

Document and rhetorical structure (W_D and W_R) take into account the location of a concept within a document (e.g., mentioned in the title) and the use of devices highlighting particular concepts (e.g., juxtaposition) in computing the overall ID score. For the intended message and informational propositions conveyed by the graphics, the weights assigned by SIGHT are incorporated into ID as W_G.

After computing the ID of each concept, we will apply Demir's (2010a) graph-based ranking algorithm to select items for the summary. This algorithm is based on PageRank (Page et al., 1999), but with several changes. Beyond centrality assessment based on relationships between concepts, it also incorporates apriori importance nodes that enable us to capture concept completeness, number of expressions, and document and rhetorical structure. More importantly from a generation perspective, Demir's algorithm iteratively selects concepts one at a time, re-ranking the remaining items by increasing the weight of related concepts and discounting redundant ones. Thus, we favor concepts that ought to be conveyed together while avoiding redundancy.

3.3 Generating a Summary

After we determine which concepts are most important as scored by ID, the next step is to decide what to say about them and express these elements as sentences. Following the generation technique of McDonald and Greenbacker (2010), the expressions observed by the parser and stored in the model are used as the "raw material" for expressing the concepts and relationships. The two most important concepts as rated in the semantic model built from the Medtronic article would be Company1 ("Medtronic") and Person1 ("Joanne Wuensch," a stock analyst). To generate a single summary sentence for this document, we should try to find some way of expressing these concepts together using the available phrasings. Since there is no direct link between these two concepts in the model (see Figure 1), none of the collected phrasings can express both concepts at the same time. Instead, we need to find a third concept that provides a semantic link between Company1 and Person1. If multiple options are available, deciding which linking concept to use becomes a microplanning problem, with the choice depending on linguistic constraints and the relative importance of the applicable linking concepts.

In this example, a reasonable selection would be TargetStockPrice1 (see Figure 1). Combining original phrasings from all three concepts (via substitution and adjunction operations on the underlying TAG trees), along with a "built-in" realization inherited by the TargetStockPrice category (a subtype of Expectation – not shown in the figure), produces a

construction resulting in this final surface form:

> *Wuensch expects a 12-month target of 62 for medical device giant Medtronic.*

Thus, we generate novel sentences, albeit with some "recycled" expressions, to form an abstractive summary of the original document.

Studies have shown that nearly 80% of human-written summary sentences are produced by a cut-and-paste technique of reusing original sentences and editing them together in novel ways (Jing and McKeown, 1999). By reusing selected short phrases ("cutting") coupled together with generalized constructions ("pasting"), we can generate abstracts similar to human-written summaries.

The set of available expressions is augmented with numerous built-in schemas for realizing common relationships such as "is-a" and "has-a," as well as realizations inherited from other conceptual categories in the hierarchy. If the knowledge base persists between documents, storing the observed expressions and making them available for later use when realizing concepts in the same category, the variety of utterances we can generate is increased. With a sufficiently rich set of expressions, the reliance on straightforward "recycling" is reduced while the amount of paraphrasing and transformation is increased, resulting in greater novelty of production. By using ongoing parser observations to support the generation process, the more the system "reads," the better it "writes."

4 Evaluation

As an intermediate evaluation, we will rate the concepts stored in a model built only from text and use this rating to select sentences containing these concepts from the original document. These sentences will be compared to another set chosen by traditional extraction methods. Human judges will be asked to determine which set of sentences best captures the most important concepts in the document. This "checkpoint" will allow us to assess how well our system identifies the most salient concepts in a text.

The summaries ultimately generated as final output by our prototype system will be evaluated against summaries written by human authors, as well as summaries created by extraction-based systems and a baseline of selecting the first few sentences. For each comparison, participants will be asked to indicate a preference for one summary over another. We propose to use preference-strength judgment experiments testing multiple dimensions of preference (e.g., accuracy, clarity, completeness). Compared to traditional rating scales, this alternative paradigm has been shown to result in better evaluator self-consistency and high inter-evaluator agreement (Belz and Kow, 2010). This allows a larger proportion of observed variations to be accounted for by the characteristics of systems undergoing evaluation, and can result in a greater number of significant differences being discovered.

Automatic evaluation, though desirable, is likely unfeasible. As human-written summaries have only about 60% agreement (Radev et al., 2002), there is no "gold standard" to compare our output against.

5 Discussion

The work proposed herein aims to advance the state-of-the-art in automatic summarization by offering a means of generating abstractive summaries from a semantic model built from the original article. By incorporating concepts obtained from non-text components (e.g., information graphics) into the semantic model, we can produce unified summaries of multimodal documents, resulting in an abstract covering the entire document, rather than one that ignores potentially important graphical content.

Acknowledgments

This work was funded in part by the National Institute on Disability and Rehabilitation Research (grant #H133G080047). The author also wishes to thank Kathleen McCoy, Sandra Carberry, and David McDonald for their collaborative support.

References

Chinatsu Aone, Mary E. Okurowski, James Gorlinsky, and Bjornar Larsen. 1999. A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automated Text Summarization*. MIT Press.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *In Proceedings*

of the ACL Workshop on Intelligent Scalable Text Summarization, pages 10–17, Madrid, July. ACL.

Anja Belz and Eric Kow. 2010. Comparing rating scales and preference judgements in language evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG 2010, pages 7–16, Trim, Ireland, July. ACL.

Sumit Bhatia, Shibamouli Lahiri, and Prasenjit Mitra. 2009. Generating synopses for document-element search. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 2003–2006, Hong Kong, November. ACM.

Sandra Carberry, Stephanie Elzer, and Seniz Demir. 2006. Information graphics: an untapped resource for digital libraries. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 581–588, Seattle, August. ACM.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI 2010)*, pages 1306–1313, Atlanta, July. AAAI.

Seniz Demir, Sandra Carberry, and Kathleen F. McCoy. 2010a. A discourse-aware graph-based content-selection framework. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG 2010, pages 17–26, Trim, Ireland, July. ACL.

Seniz Demir, David Oliver, Edward Schwartz, Stephanie Elzer, Sandra Carberry, and Kathleen F. McCoy. 2010b. Interactive SIGHT into information graphics. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility*, W4A '10, pages 16:1–16:10, Raleigh, NC, April. ACM.

Hongyan Jing and Kathleen R. McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 129–136, Berkeley, August. ACM.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 68–73, Seattle, July. ACM.

Chin-Yew Lin. 1999. Training a selection function for extraction. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, CIKM '99, pages 55–62, Kansas City, November. ACM.

Daniel C. Marcu. 1997. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. Ph.D. thesis, University of Toronto, December.

David D. McDonald and Charles F. Greenbacker. 2010. 'If you've heard it, you can say it' - towards an account of expressibility. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG 2010, pages 185–190, Trim, Ireland, July. ACL.

David D. McDonald. 1992. An efficient chart-based algorithm for partial-parsing of unrestricted texts. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pages 193–200, Trento, March. ACL.

David D. McDonald. 2000. Issues in the representation of real texts: the design of KRISP. In Lucja M. Iwańska and Stuart C. Shapiro, editors, *Natural Language Processing and Knowledge Representation*, pages 77–110. MIT Press, Cambridge, MA.

Ani Nenkova. 2006. *Understanding the process of multi-document summarization: content selection, rewrite and evaluation*. Ph.D. thesis, Columbia University, January.

Aurélie Névéol, Thomas M. Deserno, Stéfan J. Darmoni, Mark Oliver Güld, and Alan R. Aronson. 2009. Natural language processing versus content-based image analysis for medical document retrieval. *Journal of the American Society for Information Science and Technology*, 60(1):123–134.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November. Previous number: SIDL-WP-1999-0120.

Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the special issue on summarization. *Computational Linguistics*, 28(4):399–408.

Lisa F. Rau, Paul S. Jacobs, and Uri Zernik. 1989. Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing & Management*, 25(4):419 – 428.

Ulrich Reimer and Udo Hahn. 1988. Text condensation as knowledge base abstraction. In *Proceedings of the 4th Conference on Artificial Intelligence Applications*, CAIA '88, pages 338–344, San Diego, March. IEEE.

Michael J. Witbrock and Vibhu O. Mittal. 1999. Ultra-summarization: a statistical approach to generating highly condensed non-extractive summaries. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 315–316, Berkeley, August. ACM.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd Int'l Conference on Computational Linguistics*, COLING '08, pages 985–992, Manchester, August. ACL.

Sentiment Analysis of Citations using Sentence Structure-Based Features

Awais Athar
University of Cambridge
Computer Laboratory
15 JJ Thompson Avenue
Cambridge, CB3 0FD, U.K.
awais.athar@cl.cam.ac.uk

Abstract

Sentiment analysis of citations in scientific papers and articles is a new and interesting problem due to the many linguistic differences between scientific texts and other genres. In this paper, we focus on the problem of automatic identification of positive and negative sentiment polarity in citations to scientific papers. Using a newly constructed annotated citation sentiment corpus, we explore the effectiveness of existing and novel features, including n-grams, specialised science-specific lexical features, dependency relations, sentence splitting and negation features. Our results show that 3-grams and dependencies perform best in this task; they outperform the sentence splitting, science lexicon and negation based features.

1 Introduction

Sentiment analysis is the task of identifying positive and negative opinions, sentiments, emotions and attitudes expressed in text. Although there has been in the past few years a growing interest in this field for different text genres such as newspaper text, reviews and narrative text, relatively less emphasis has been placed on extraction of opinions from scientific literature, more specifically, citations. Analysis of citation sentiment would open up many exciting new applications in bibliographic search and in bibliometrics, i.e., the automatic evaluation the influence and impact of individuals and journals via citations.

Existing bibliometric measures like H-Index (Hirsch, 2005) and adapted graph ranking algorithms like PageRank (Radev et al., 2009) treat all citations as equal. However, Bonzi (1982) argued that if a cited work is criticised, it should consequently carry lower or even negative weight for bibliometric measures. Automatic citation sentiment detection is a prerequisite for such a treatment.

Moreover, citation sentiment detection can also help researchers during search, by detecting problems with a particular approach. It can be used as a first step to scientific summarisation, enable users to recognise unaddressed issues and possible gaps in the current research, and thus help them set their research directions.

For other genres a rich literature on sentiment detection exists and researchers have used a number of features such as n-grams, presence of adjectives, adverbs and other parts-of-speech (POS), negation, grammatical and dependency relations as well as specialised lexicons in order to detect sentiments from phrases, words, sentences and documents. State-of-the-art systems report around 85-90% accuracy for different genres of text (Nakagawa et al., 2010; Yessenalina et al., 2010; Täckström and McDonald, 2011).

Given such good results, one might think that a sentence-based sentiment detection system trained on a different genre could be used equally well to classify citations. We argue that this might not be the case; our citation sentiment recogniser uses specialised training data and tests the performance of specialised features against current state-of-the-art features. The reasons for this are based on the following observations:

- Sentiment in citations is often hidden. This might

Proceedings of the ACL-HLT 2011 Student Session, pages 81–87,
Portland, OR, USA 19-24 June 2011.

be because of the general strategy to avoid overt criticism due to the sociological aspect of citing (MacRoberts and MacRoberts, 1984; Thompson and Yiyun, 1991). Ziman (1968) states that many works are cited out of "politeness, policy or piety". Negative sentiment, while still present and detectable for humans, is expressed in subtle ways and might be hedged, especially when it cannot be quantitatively justified (Hyland, 1995).

*While SCL has been successfully applied to POS tagging and Sentiment Analysis (Blitzer et al., 2006), its effectiveness for parsing was **rather unexplored**.*

- Citation sentences are often neutral with respect to sentiment, either because they describe an algorithm, approach or methodology objectively, or because they are used to support a fact or statement.

  There are five different IBM translation models (Brown et al. , 1993).

  This gives rise to a far higher proportion of objective sentences than in other genres.

- Negative polarity is often expressed in contrastive terms, e.g. in evaluation sections. Although the sentiment is indirect in these cases, its negativity is implied by the fact that the authors' own work is clearly evaluated positively in comparison.

  *This method was shown to **outperform** the class based model proposed in (Brown et al., 1992) . . .*

- There is also much variation between scientific texts and other genres concerning the lexical items chosen to convey sentiment. Sentiment carrying science-specific terms exist and are relatively frequent, which motivates the use of a sentiment lexicon specialised to science.

  *Similarity-based smoothing (Dagan, Lee, and Pereira 1999) provides an **intuitively appealing** approach to language modeling.*

- Technical terms play a large role overall in scientific text (Justeson and Katz, 1995). Some of these carry sentiment as well.

  *Current **state of the art** machine translation systems (Och, 2003) use phrasal (n-gram) features . . .*

  For this reason, using higher order n-grams might prove to be useful in sentiment detection.

- The scope of influence of citations varies widely from a single clause (as in the example below) to several paragraphs:

  *As reported in Table 3, small increases in METEOR (**Banerjee and Lavie, 2005**), BLEU (Papineni et al., 2002) and NIST scores (Doddington, 2002) suggest that . . .*

  This affects lexical features directly since there could be "sentiment overlap" associated with neighbouring citations. Ritchie et al. (2008) showed that assuming larger citation scopes has a positive effect in retrieval. We will test the opposite direction here, i.e., we assume short scopes and use a parser to split sentences, so that the features associated with the clauses not directly connected to the citation are disregarded.

We created a new sentiment-annotated corpus of scientific text in the form of a sentence-based collection of over 8700 citations. Our experiments use a supervised classifier with the state-of-the-art features from the literature, as well as new features based on the observations above. Our results show that the most successful feature combination includes dependency features and n-grams longer than for other genres ($n = 3$), but the assumption of a smaller scope (sentence splitting) decreased results.

2 Training and Test Corpus

We manually annotated 8736 citations from 310 research papers taken from the ACL Anthology (Bird et al., 2008). The citation summary data from the ACL Anthology Network[1] (Radev et al., 2009) was used. We identified the actual text of the citations by regular expressions and replaced it with a special token $<CIT>$ in order to remove any lexical bias associated with proper names of researchers. We labelled each sentence as positive, negative or objective, and separated 1472 citations for development and training. The rest were used as the test set containing 244 negative, 743 positive and 6277 objective citations. Thus our dataset is heavily skewed, with subjective citations accounting for only around 14% of the corpus.

[1] http://www.aclweb.org

3 Features

We represent each citation as a feature set in a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) framework which has been shown to produce good results for sentiment classification (Pang et al., 2002). The corpus is processed using WEKA (Hall et al., 2008) and the Weka LibSVM library (EL-Manzalawy and Honavar, 2005; Chang and Lin, 2001) with the following features.

3.1 Word Level Features

In accordance with Pang et al. (2002), we use unigrams and bigrams as features and also add 3-grams as new features to capture longer technical terms. POS tags are also included using two approaches: attaching the tag to the word by a delimiter, and appending all tags at the end of the sentence. This may help in distinguishing between homonyms with different POS tags and signalling the presence of adjectives (e.g., JJ) respectively. Name of the primary author of the cited paper is also used as a feature.

A science-specific sentiment lexicon is also added to the feature set. This lexicon consists of 83 polar phrases which have been manually extracted from the development set of 736 citations. Some of the most frequently occurring polar phrases in this set consists of adjectives such as *efficient*, *popular*, *successful*, *state-of-the-art* and *effective*.

3.2 Contextual Polarity Features

Features previously found to be useful for detecting phrase-level contextual polarity (Wilson et al., 2009) are also included. Since the task at hand is sentence-based, we use only the sentence-based features from the literature e.g., presence of subjectivity clues which have been compiled from several sources[2] along with the number of adjectives, adverbs, pronouns, modals and cardinals.

To handle negation, we include the count of negation phrases found within the citation sentence. Similarly, the number of valance shifters (Polanyi and Zaenen, 2006) in the sentence are also used. The polarity shifter and negation phrase lists have been taken from the OpinionFinder system (Wilson et al., 2005).

[2] Available for download at http://www.cs.pitt.edu/mpqa/

3.3 Sentence Structure Based Features

We explore three different feature sets which focus on the lexical and grammatical structure of a sentence and have not been explored previously for the task of sentiment analysis of scientific text.

3.3.1 Dependency Structures

The first set of these features include typed dependency structures (de Marneffe and Manning, 2008) which describe the grammatical relationships between words. We aim to capture the long distance relationships between words. For instance in the sentence below, the relationship between *results* and *competitive* will be missed by trigrams but the dependency representation captures it in a single feature `nsubj_competitive_results`.

<CIT> showed that the results for French-English were competitive to state-of-the-art alignment systems.

A variation we experimented with, but gave up on as it did not show any improvements, concerns backing-off the dependent and governor to their POS tags (Joshi and Penstein-Rosé, 2009).

3.3.2 Sentence Splitting

Removing irrelevant polar phrases around a citation might improve results. For this purpose, we split each sentence by trimming its parse tree. Walking from the citation node (*<CIT>*) towards the root, we select the subtree rooted at the first sentence node (*S*) and ignore the rest. For example, in Figure 1, the cited paper is not included in the scope of the discarded polar phrase *significant improvements*.

```
(ROOT
  (S
    (NP (DT This) (JJ new) (NN model))
    (VP (VBZ leads)
      (PP (TO to)
        (NP
          (NP (JJ significant) (NNS improvements))
          (PP (IN in)
            (NP (NN MT) (NN quality)))))
      (SBAR (IN as)
        (S
          (VP (VBN measured)
            (PP (IN by)
              (NP (NNP BLEU) (NNP <CIT>)))))))
    (. .)))
```

Figure 1: An example of parse tree trimming

3.3.3 Negation

Dependencies and parse trees attach negation nodes, such as *not*, to the clause subtree and this shows no interaction with other nodes with respect to valence shifting. To handle this effect, we take a simple window-based inversion approach. All words inside a k-word window of any negation term are suffixed with a token *_neg* to distinguish them from their non-polar versions. For example, a 2-word negation window inverts the polarity of the positive phrase *work well* in the sentence below.

Turney's method did not work_neg well_neg although they reported 80% accuracy in <CIT>.

The negation term list has been taken from the OpinionFinder system. Khan (2007) has shown that this approach produces results comparable to grammatical relations based negation models.

4 Results

Because of our skewed dataset, we report both the macro-F and the micro-F scores using 10-fold cross-validation (Lewis, 1991). The bold values in Table 1 show the best results.

Features	macro-F	micro-F
1 grams	0.581	0.863
1-2 grams	0.592	0.864
1-3 grams	0.597	0.862
$''$ + POS	0.535	0.859
$''$ + POS (tokenised)	0.596	0.859
$''$ + scilex	0.597	0.860
$''$ + wlev	0.535	0.859
$''$ + cpol	0.418	0.859
$''$ + dep	0.760	0.897
$''$ + dep + split + neg	0.683	0.872
$''$ + dep + split	0.642	0.866
$''$ + dep + neg	**0.764**	**0.898**

Table 1: Results using science lexicon (scilex), contextual polarity (cpol), dependencies (dep), negation (neg), sentence splitting (split) and word-level (wlev) features.

The selection of the features is on the basis of improvements over a baseline of 1-3 grams i.e. if a feature (e.g. scilex) did not shown any improvement, it is has been excluded from the subsequent experiments.

The results show that contextual polarity features do not work well on citation text. Adding a science-specific lexicon does not help either. This may indicate that n-grams are sufficient to capture discriminating lexical structures. We find that word level and contextual polarity features are surpassed by dependency features. Sentence splitting does not help, possibly due to longer citation scope. Adding a negation window (k=15) improves the performance but the improvement was not found to be statistically significant. This might be due to skewed class distribution and a larger dataset may prove to be useful.

5 Related Work

While different schemes have been proposed for annotating citations according to their function (Spiegel-Rösing, 1977; Nanba and Okumura, 1999; Garzone and Mercer, 2000), there have been no attempts on citation sentiment detection in a large corpus.

Teufel et al. (2006) worked on a 2829 sentence citation corpus using a 12-class classification scheme. However, this corpus has been annotated for the task of determining the author's reason for citing a given paper and is thus built on top of sentiment of citation. It considers usage, modification and similarity with a cited paper as positive even when there is no sentiment attributed to it. Moreover, contrast between two cited methods (CoCoXY) is categorized as objective in the annotation scheme even if the text indicates that one method performs better than the other. For example, the sentence below talks about a positive attribute but is marked as neutral in the scheme.

Lexical transducers are more efficient for analysis and generation than the classical two-level systems (Koskenniemi,1983) because ...

Using this corpus is thus more likely to lead to inconsistent representation of sentiment in any system which relies on lexical features. Teufel et al. (2006) group the 12 categories into 3 in an attempt to perform a rough approximation of sentiment analysis over the classifications and report a 0.710 macro-F score. Unfortunately, we have ac-

cess to only a subset[3] of this citation function corpus. We have extracted 1-3 grams, dependencies and negation features from the reduced citation function dataset and used them in our system with 10-fold cross-validation. This results in an improved macro-F score of 0.797 for the subset. This shows that our system is comparable to Teufel et al. (2006). When this subset is used to test the system trained on our newly annotated corpus, a low macro-F score of 0.484 is achieved. This indicates that there is a mismatch in the annotated class labels. Therefore, we can infer that citation sentiment classification is different from citation function classification.

Other approaches to citation annotation and classification include Wilbur et al. (2006) who annotated a small 101 sentence corpus on focus, polarity, certainty, evidence and directionality. Piao et al. (2007) proposed a system to attach sentiment information to the citation links between biomedical papers.

Different dependency relations have been explored by Dave et al. (2003), Wilson et al. (2004) and Ng et al. (2006) for sentiment detection. Nakagawa et al. (2010) report that using dependencies on conditional random fields with lexicon based polarity reversal results in improvements over n-grams for news and reviews corpora.

A common approach is to use a sentiment labelled lexicon to score sentences (Hatzivassiloglou and McKeown, 1997; Turney, 2002; Yu and Hatzivassiloglou, 2003). Research suggests that creating a general sentiment classifier is a difficult task and existing approaches are highly topic dependent (Engström, 2004; Gamon and Aue, 2005; Blitzer et al., 2007).

6 Conclusion

In this paper, we focus on automatic identification of sentiment polarity in citations. Using a newly constructed annotated citation sentiment corpus, we examine the effectiveness of existing and novel features, including n-grams, scientific lexicon, dependency relations and sentence splitting. Our results show that 3-grams and dependencies perform best in this task; they outperform the scientific lexicon and the sentence splitting features. Future directions include trying to improve the performance by modelling negations using a more sophisticated approach. New techniques for detection of the negation scope such as the one proposed by Councill et al. (2010) might also be helpful in citations. Exploring longer citation scopes by including citation contexts might also improve citation sentiment detection.

[3]This subset contains 591 positive, 59 negative and 1259 objective citations.

References

S. Bird, R. Dale, B.J. Dorr, B. Gibson, M.T. Joseph, M.Y. Kan, D. Lee, B. Powley, D.R. Radev, and Y.F. Tan. 2008. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proc. of the 6th International Conference on Language Resources and Evaluation Conference (LREC08)*, pages 1755–1759. Citeseer.

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 45, page 440.

S. Bonzi. 1982. Characteristics of a literature as predictors of relatedness between cited and citing works. *Journal of the American Society for Information Science*, 33(4):208–216.

C.C. Chang and C.J. Lin. 2001. LIBSVM: a library for support vector machines, 2001. *Software available at* `http://www.csie.ntu.edu.tw/cjlin/libsvm`.

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

I.G. Councill, R. McDonald, and L. Velikovich. 2010. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 51–59. Association for Computational Linguistics.

K. Dave, S. Lawrence, and D.M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.

M.C. de Marneffe and C.D. Manning. 2008. The Stanford typed dependencies representation. In *COLING*, pages 1–8. Association for Computational Linguistics.

Y. EL-Manzalawy and V. Honavar, 2005. *WLSVM: Integrating LibSVM into Weka Environment*. Software available at `http://www.cs.iastate.edu/~yasser/wlsvm`.

C. Engström. 2004. Topic dependence in sentiment classification. *Unpublished MPhil Dissertation. University of Cambridge*.

M. Gamon and A. Aue. 2005. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*, pages 57–64. Association for Computational Linguistics.

M. Garzone and R. Mercer. 2000. Towards an automated citation classifier. *Advances in Artificial Intelligence*, pages 337–346.

D. Hall, D. Jurafsky, and C.D. Manning. 2008. Studying the history of ideas using topic models. In *EMNLP*, pages 363–371.

V. Hatzivassiloglou and K.R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of EACL*, pages 174–181. Association for Computational Linguistics.

J.E. Hirsch. 2005. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569.

K. Hyland. 1995. The Author in the Text: Hedging Scientific Writing. *Hong Kong papers in linguistics and language teaching*, 18:11.

M. Joshi and C. Penstein-Rosé. 2009. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 313–316. Association for Computational Linguistics.

J.S. Justeson and S.M. Katz. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1(01):9–27.

S. Khan. 2007. *Negation and Antonymy in Sentiment Classification*. Ph.D. thesis, Computer Lab, University of Cambridge.

D.D. Lewis. 1991. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318.

M.H. MacRoberts and B.R. MacRoberts. 1984. The negational reference: Or the art of dissembling. *Social Studies of Science*, 14(1):91–94.

T. Nakagawa, K. Inui, and S. Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *NAACL HLT*, pages 786–794. Association for Computational Linguistics.

H. Nanba and M. Okumura. 1999. Towards multi-paper summarization using reference information. In *IJCAI*, volume 16, pages 926–931. Citeseer.

V. Ng, S. Dasgupta, and SM Arifin. 2006. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618. Association for Computational Linguistics.

B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP*, pages 79–86. Association for Computational Linguistics.

S. Piao, S. Ananiadou, Y. Tsuruoka, Y. Sasaki, and J. McNaught. 2007. Mining opinion polarity relations of citations. In *International Workshop on Computational Semantics (IWCS)*, pages 366–371. Citeseer.

L. Polanyi and A. Zaenen. 2006. Contextual valence shifters. *Computing attitude and affect in text: Theory and applications*, pages 1–10.

D.R. Radev, M.T. Joseph, B. Gibson, and P. Muthukrishnan. 2009. A Bibliometric and Network Analysis of the field of Computational Linguistics. *Journal of the American Society for Information Science and Technology*, 1001:48109–1092.

A. Ritchie, S. Robertson, and S. Teufel. 2008. Comparing citation contexts for information retrieval. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pages 213–222. ACM.

I. Spiegel-Rösing. 1977. Science studies: Bibliometric and content analysis. *Social Studies of Science*, 7(1):97–113.

O. Täckström and R. McDonald. 2011. Discovering fine-grained sentiment with latent variable structured prediction models. In *Proceedings of the ECIR*.

S. Teufel, A. Siddharthan, and D. Tidhar. 2006. Automatic classification of citation function. In *EMNLP*, pages 103–110. Association for Computational Linguistics.

G. Thompson and Y. Yiyun. 1991. Evaluation in the reporting verbs used in academic papers. *Applied linguistics*, 12(4):365.

P.D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424. Association for Computational Linguistics.

W.J. Wilbur, A. Rzhetsky, and H. Shatkay. 2006. New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC bioinformatics*, 7(1):356.

T. Wilson, J. Wiebe, and R. Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of the National Conference on Artificial Intelligence*, pages 761–769. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP*, pages 347–354. Association for Computational Linguistics.

T. Wilson, J. Wiebe, and P. Hoffmann. 2009. Recognizing Contextual Polarity: an exploration of features for

phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

A. Yessenalina, Y. Yue, and C. Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of EMNLP*, pages 1046–1056, Cambridge, MA, October. Association for Computational Linguistics.

H. Yu and V. Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP*, pages 129–136. Association for Computational Linguistics.

J.M. Ziman. 1968. *Public Knowledge: An essay concerning the social dimension of science*. Cambridge Univ. Press, College Station, Texas.

Combining Indicators of Allophony

Luc Boruta
Univ. Paris Diderot, Sorbonne Paris Cité, ALPAGE, UMR-I 001 INRIA, F-75205, Paris, France
LSCP, Département d'Études Cognitives, École Normale Supérieure, F-75005, Paris, France
luc.boruta@inria.fr

Abstract

Allophonic rules are responsible for the great variety in phoneme realizations. Infants can not reliably infer abstract word representations without knowledge of their native allophonic grammar. We explore the hypothesis that some properties of infants' input, referred to as indicators, are correlated with allophony. First, we provide an extensive evaluation of individual indicators that rely on distributional or lexical information. Then, we present a first evaluation of the combination of indicators of different types, considering both logical and numerical combinations schemes. Though distributional and lexical indicators are not redundant, straightforward combinations do not outperform individual indicators.

1 Introduction

Though the phonemic inventory of a language is typically small, phonetic and phonological processes yield manifold variants[1] for each phoneme. Words too are affected by this variability, yielding different realizations for a given underlying form. Allophonic rules relate phonemes to their variants, expressing the contexts in which the latter occur. We are interested in describing procedures by which infants, learning their native allophonic grammar, could reduce the variation and recover words. Combining insights from both computational and behavioral studies, we endorse the hypothesis that infants are good distributional learners (Maye et al., 2002; Saffran et al., 1996) and that they may 'bootstrap' into language tracking statistical regularities in the signal.

We seek to identify which features of infants' input are most reliable for learning allophonic rules. A few indicators, based on distributional (Peperkamp et al., 2006) and lexical (Martin et al., submitted) information, have been described and validated *in silico*.[2] Yet, other aspects have barely been addressed, e.g. the question of whether or not these indicators capture different aspects of allophony and, if so, which combination scheme yields better results.

We present an extensive evaluation of individual indicators and, based on theoretical and empirical desiderata, we outline a more comprehensive framework to model the acquisition of allophonic rules.

2 Indicators of allophony

We build upon Peperkamp et al.'s framework: the task is to induce a two-class classifier deciding, for every possible pair of segments, whether or not they realize the same phoneme. Discrimination relies on indicators, i.e. linguistic properties which are correlated with allophony. As a model of language acquisition, this classifier is induced without supervision.

In line with previous studies, we assume that infants are able to segment the continuous stream of acoustic input into a sequence of discrete segments, and that they quantize each of these segments into one of a finite number of phonetic categories. Quantization is a necessary assumption for the framework to apply. However, the larger the set of phonetic categories, the closer we get to recent 'single-stage' approaches (e.g. work by Dillon et al., in preparation) where phonological categories are acquired directly from raw infant-directed speech.

[1] We use *allophony* as an umbrella term for the continuum ranging from typical allophones to mere coarticulatory variants.

[2] See also the work of Dautriche (2009) on acoustic indicators of allophony, albeit using adult-directed speech.

Proceedings of the ACL-HLT 2011 Student Session, pages 88–93,
Portland, OR, USA 19-24 June 2011.

2.1 Distributional indicators

Complementary distribution is a ubiquitous criterion for the discovery of phonemes. If two segments occur in mutually exclusive contexts, the two may be realizations of the same phoneme.

Bearing in mind that the signal may be noisy, Peperkamp et al. (2006) looked for segments in near-complementary distributions. Using the symmetrised Kullback–Leibler divergence (henceforth KL), they compared the probability distributions of how often the contexts of each segment occur. In a follow-up study, Le Calvez (2007) compared KL to other indicators, namely the Jensen–Shannon divergence (JS) and the Bhattacharyya coefficient (BC).[3]

2.2 Lexical indicators

Adjacent segments can condition the realization of a word's initial and final phonemes. If two words only differ by their initial or final segments, these segments may be realizations of the same phoneme. Instantiating the general concept of functional load (Hockett, 1955), lexical indicators gauge the degree of contrast in the lexicon between two segments.

Using the simplest expression of functional load, Martin et al. (submitted) defined a Boolean-valued indicator, FL, satisfied by a single pair of minimally different words. As a result, FL is sensitive to noise. We define a finer-grained variant, FL*, which tallies the number of such pairs. Moreover, as words get longer, it becomes increasingly unlikely that such word pairs occur by chance. Thus, for any such pair, FL* is incremented by the length of those words.

We also propose an information-theoretic lexical indicator, HFL, based on Hockett's definition of functional load. HFL accounts for the fraction of information content, represented by the language's word entropy, that is lost when the opposition between two segments is neutralized. The 'broken typewriter' function used for neutralization guarantees that values lie in $[0, 1]$ (Coolen et al., 2005).

3 Corpora and experimental setup

In the absence of phonetic transcriptions of infant-directed speech, and as the number of allophones infants must learn is unknown (if assessable at all), we use Boruta et al.'s (submitted) corpora. They created a range of possible inputs, applying artificial allophonic grammars[4] of different sizes (Boruta, 2011) to the now-standard CHILDES 'Brent/Ratner' corpus of English (Brent and Cartwright, 1996). We quantify the amount of variation in a corpus by its allophonic complexity, i.e. the ratio of the number of phones to the number of phonemes in the language.

Lexical indicators require an ancillary procedure yielding a lexicon. Martin et al. approximated a lexicon by a list of frequent n-grams. Here, the lexicon is induced from the output of an explicit word segmentation model, viz. Venkataraman's incremental (2001) model, using the unsegmented phonetic corpora as the input. Though, obviously, infants can not access it, we use the lexicon derived from the CHILDES orthographic transcripts for reference.

4 Indicators' discriminant power

As the aforementioned indicators have been evaluated using various languages, allophonic grammars and measures, we present a unified evaluation, conducted using Sing et al.'s (2005) ROCR package.

4.1 Evaluation

Non-Boolean indicators require a threshold at and above which pairs are classified as allophonic. We evaluate indicators across all possible discrimination thresholds, reporting the area under the ROC curve (henceforth AUC). Equivalent to Martin et al.'s ρ, values lie in $[0, 1]$ and are equal to the probability that a randomly drawn allophonic pair will score higher than a randomly drawn non-allophonic pair; .5 thus indicates random prediction.

Moreover, we evaluate indicators' misclassifications at the discrimination threshold maximizing Matthews' (1975) correlation coefficient: let α, β, γ and δ be, respectively, the number of false positives, false negatives, true positives and true negatives, $\text{MCC} = (\gamma\delta - \alpha\beta)/\sqrt{(\alpha+\gamma)(\beta+\gamma)(\alpha+\delta)(\beta+\delta)}$. Values of 1, 0 and -1 indicate perfect, random and inverse prediction, respectively. This coefficient is more appropriate than the accuracy or the F-measure

[3] As for the actual computations, we use the same definitions as Le Calvez (2007) except that, as BC increases when distributions overlap and $0 \leq \text{BC} \leq 1$, we actually use $1 - \text{BC}$.

[4] Because all allophonic rules implemented in the corpora are of the type $p \rightarrow a \;/\; \_\; c$, FL and FL* only look for words minimally differing by their last segments.

when, as here, the true classes have very different sizes.[5] Using this optimal, MCC-maximizing threshold, we report the maximal MCC and, as percentages, the accuracy (Acc), the false positive rate (FPR) and the false negative rate (FNR).

4.2 Results and discussion

Indicators' AUC corroborate previous results for distributional indicators: they perform almost identically and do not accommodate high allophonic complexities at which they perform below chance (Figure 1.a) because, as suggested by Martin et al., every segment has an extremely narrow distribution and complementary distribution is the rule rather than the exception. By contrast, all three lexical indicators are much more robust even if, as predicted, FL's coarseness impedes its discriminant power (Figure 1.b).[6] The reason why FL* outperforms HFL may be due to the very definition of HFL's broken typewriter function: as the segments, e.g. $\{x, y\}$, are collapsed into a single symbol, the indicator captures not only minimal alternations like $wx \sim wy$, but also word pairs such as $xy \sim yx$.

AUC curves suggest that, for each type, indicators converge at medium allophonic complexity. Thus, misclassification scores are reported in Table 1 only at low (2 allophones/phoneme) and medium (9) complexities. Previous observations are confirmed by MCC and accuracy values: though all indicators are positively correlated with the underlying allophonic relation, correlation is stronger for lexical indicators. Surprisingly, zero FPR values are observed for some lexical indicators, meaning that they make no false alarms and, as a consequence, that all errors are caused by missed allophonic pairs.

5 Indicators' redundancy

None of the indicators we benchmarked in the previous section makes a perfect discrimination between allophonic and non-allophonic pairs of segments.

[5] If p phonemes have on average a allophones, out of the $pa(pa-1)/2$ possible pairs, only $pa(a-1)/2$ are allophonic, and a dummy indicator that rejects all pairs achieves a constant accuracy of $1-(a-1)/(pa-1)$, which is greater than 98% for any of our corpora. Besides, the computation of precision, recall and the F-measure do not take true negatives into account.

[6] These indicators perform similarly using the orthographic lexicon: we only report AUC for FL* (referred to as oFL*), as it gives the upper bound on lexical indicators' performance.

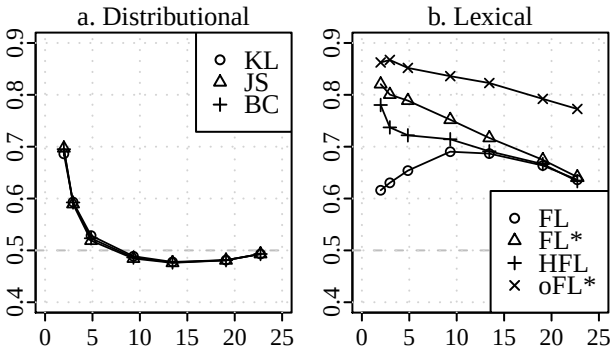

Figure 1: Indicators' AUC as a function of allophonic complexity. The dashed line indicates random prediction.

	2 allophones/phoneme				9 allophones/phoneme			
	MCC	Acc	FPR	FNR	MCC	Acc	FPR	FNR
KL	.095	*88.2*	11.3	58.5	.017	*90.7*	07.8	88.8
JS	.097	*86.4*	13.1	53.7	.014	*93.3*	05.1	93.0
BC	.097	*86.8*	12.8	54.4	.016	*89.9*	08.6	88.1
FL	.048	*37.3*	63.2	**13.6**	.116	*73.1*	26.8	**35.2**
FL*	**.564**	**99.3**	**00.0**	67.3	**.563**	**98.6**	**00.4**	53.0
HFL	.301	99.1	**00.0**	87.8	.125	*94.1*	04.5	78.7

Table 1: Indicators' performance at low and medium complexities, using the MCC-maximizing thresholds. Boldface indicates the best value. Italics indicate accuracies below that of a dummy indicator rejecting all pairs.

Yet, if some segment pairs are misclassified by one but not all (types of) indicators, a suitable combination should outperform individual indicators. In other words, combining indicators may yield better results only if, individually, indicators capture different subsets of the underlying allophonic relation.

5.1 Evaluation

To get a straightforward estimation of redundancy, we compute the Jaccard index between each indicator's set of misclassified pairs: let D and L be sets containing, respectively, a distributional and a lexical indicator's errors, $J(D, L) = |D \cap L|/|D \cup L|$. Values lie in $[0, 1]$ and the lower the index, the more promising the combination. To distinguish false positives from false negatives, we compute two Jaccard indices for each possible combination.

5.2 Results and discussion

Jaccard indices, reported in Table 2, emphasize the distinction between false positives and false negatives. False negatives have rather high indices: most

allophonic pairs that are not captured by distributional indicators are not captured either by lexical indicators, and *vice versa*. By contrast, there is little or no redundancy in false positives, even at medium allophonic complexity: though random pairs can be incorrectly classified as allophonic, the error is unlikely to recur across all types of indicators.

It is also worth noting that though JS performs slightly better than KL and BC, the exact nature of the distributional indicator seems to have little influence on the performance of the combination.

6 Combining indicators

As distributional and lexical indicators are not completely redundant, combining them is a natural extension. However, not all conceivable combination schemes are appropriate for our task. We present our choices in terms of Marr's (1982) levels of analysis.

At the computational level, a combination scheme can be either disjunctive or conjunctive, i.e. each indicator can be either sufficient or (only) necessary. Aforementioned indicators were designed as necessary but not sufficient correlates of phonemehood. For instance, while a phoneme's allophones have complementary distributions, not all segments that have complementary distributions are allophones of a single phoneme. Therefore, we favor a conjunctive scheme,[7] even if this conflicts with abovementioned results: most errors are due to missed allophonic pairs but a conjunctive scheme, where every indicator must be satisfied, is likely to increase misses.

At the algorithmic level, a combination scheme can be either logical or numerical. A logical scheme uses a logical connective to join indicators' Boolean decisions, typically by conjunction according to our previous decision. By contrast, a numerical scheme tries to approximate interactions between indicators' values, merging them using any monotone increasing function; discrimination then relies on a single threshold. In practical terms, we use multiplication as a numerical counterpart of conjunction.

6.1 Evaluation

Setting aside the following minor adjustments, we use the same protocol as for individual indicators.

[7]This generalizes Martin et al.'s attempt at combination: they used FL as a high-pass lexical filter prior to the use of KL.

		2 allo./phon.		9 allo./phon.	
		FP	FN	FP	FN
KL	FL	.096	**.071**	.113	.359
JS	FL	.113	.076	.071	**.355**
BC	FL	.110	.075	.118	.358
KL	FL*	**.000**	.595	**.008**	.520
JS	FL*	**.000**	.548	**.005**	.525
BC	FL*	**.000**	.556	**.007**	.517
KL	HFL	**.000**	.667	.087	.788
JS	HFL	**.000**	.612	.033	.781
BC	HFL	**.000**	.620	.089	.787

Table 2: Indicators' redundancy at low and medium allophonic complexities, estimated by the Jaccard indices between their false positives (FP) and false negatives (FN). Boldface indicates the best value.

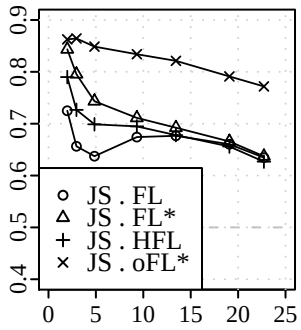

Figure 2: Indicators' AUC as a function of allophonic complexity, for the multiplicative combination scheme. The dashed line indicates random prediction.

Logical combinations require one discrimination threshold per combined indicator. As it facilitates comparison with previous results, we report performance at the thresholds maximizing the MCC of individual indicators (rather than at the thresholds maximizing the combined MCC) .

Numerical combinations are sensitive to differences in indicators' magnitudes. Equal contribution of all indicators may or may not be a desirable property, but in the absence of *a priori* knowledge of indicators' relative weights, each indicator's values were standardized so that they lie in $[0, 1]$, shifting the minimum to zero and rescaling by the range.

6.2 Results and discussion

It is worth noting that, while the performance of combined indicators is still good (Table 3), it is less satisfactory than that of the best individual indicators. Moreover, even if misclassification scores

		Logical combination: conjunction								Numerical combination: multiplication							
		2 allophones/phoneme				9 allophones/phoneme				2 allophones/phoneme				9 allophones/phoneme			
		MCC	Acc	FPR	FNR	MCC	Acc	FPR	FNR	MCC	Acc	FPR	FNR	MCC	Acc	FPR	FNR
KL	FL	.104	*92.9*	06.5	67.3	.037	*94.7*	03.6	91.3	.104	*92.9*	06.5	67.3	.116	*73.1*	26.7	**35.2**
JS	FL	.109	*91.7*	07.8	**62.6**	.032	*96.2*	02.1	94.6	.110	*91.5*	07.9	**61.9**	.116	*73.1*	26.7	**35.2**
BC	FL	.109	*91.9*	07.5	63.3	.038	*94.5*	03.9	**90.8**	.109	*92.8*	06.6	66.0	.116	*73.1*	26.7	**35.2**
KL	FL*	.457	**99.2**	**00.0**	78.9	.207	**98.2**	00.1	93.3	.526	**99.3**	**00.0**	71.4	.371	**98.4**	**00.1**	81.6
JS	FL*	**.465**	**99.2**	**00.0**	78.2	.153	**98.2**	**00.0**	95.7	**.548**	**99.3**	**00.0**	66.0	**.393**	**98.4**	00.2	78.3
BC	FL*	**.465**	**99.2**	**00.0**	78.2	**.211**	**98.2**	00.1	93.0	.535	**99.3**	**00.0**	68.7	.388	**98.4**	**00.1**	79.0
KL	HFL	.348	99.1	**00.0**	87.8	.078	*97.0*	01.3	93.5	.363	99.1	**00.0**	84.4	.117	*90.3*	08.4	73.7
JS	HFL	.348	99.1	**00.0**	87.8	.068	*97.9*	00.3	96.5	.359	99.1	00.1	83.7	.119	*90.4*	08.4	73.9
BC	HFL	.348	99.1	**00.0**	87.8	.077	*96.9*	01.4	93.2	.361	99.1	**00.0**	85.7	.119	*90.3*	08.4	73.5

Table 3: Performance of combined distributional and lexical indicators, at low and medium allophonic complexity. Boldface indicates the best value. Italics indicate accuracies below that of a dummy indicator rejecting all pairs.

show that conjoined and multiplied indicators perform similarly, disparities emerge at medium allophonic complexity: while multiplication yields better MCC and FNR, conjunction yields better accuracy and FPR. In that regard, observing FPR values of zero is quite satisfactory from the point of view of language acquisition, as processing two segments as realizations of a single phoneme (while they are not) may lead to the confusion of true minimal pairs of words. Indeed, at a higher level, learning allophonic rules allows the infant to reduce the size of its emerging lexicon, factoring out allophonic realizations for each underlying word form.

Furthermore, AUC curves for the multiplicative scheme (Figure 2),[8] most notably FL's, suggest that distributional indicators' contribution to the combinations appears to be rather negative, except at very low allophonic complexities. One explanation (yet to be tested experimentally) would be that they come into play later in the learning process, once part of allophony has been reduced using other indicators.

7 Conclusion

We presented an evaluation of distributional and lexical indicators of allophony. Although they all perform well at low allophonic complexities, misclassifications increase, more or less seriously, when the average number of allophones per phoneme increases. We also presented a first evaluation of the combination of indicators, and found no significant difference between the two combination schemes we defined. Unfortunately, none of the combinations we tested outperforms individual indicators.

For comparability with previous studies, we only considered combination schemes requiring no modification in the definition of the task; however, learning allophonic pairs becomes unnatural when phonemes can have more than two realizations. Embedding each indicator's segment-to-segment (dis)similarities in a multidimensional space, for example, would enable the use of clustering techniques where minimally distant points would be analyzed as allophones of a single phoneme.

Thus far, segments have been nothing but abstract symbols and, for example, the task at hand is as hard for [a] $\sim$ [ḁ] as it is for [ɰ] $\sim$ [k]. However, not only do allophones of a given phoneme tend to be acoustically similar, but acoustic differences may be more salient and/or available earlier to the infant than complementary distributions or minimally differing words. Therefore, the main extension towards a comprehensive model of the acquisition of allophonic rules would be to include acoustic indicators.

Acknowledgments

This work was supported by a graduate fellowship from the French Ministery of Research. We thank Benoît Crabbé, Emmanuel Dupoux and Sharon Peperkamp for helpful comments and discussion.

[8] We do not report a threshold-free evaluation for the logical scheme. As it requires the estimation of the volume under a surface, comparison between schemes becomes difficult. Moreover, as the exact definition of the distributional indicator does not affect the results, we only plot combinations with JS.

References

Luc Boruta, Sharon Peperkamp, Benoît Crabbé, and Emmanuel Dupoux. Submitted. Testing the robustness of online word segmentation: effects of linguistic diversity and phonetic variation.

Luc Boruta. 2011. A note on the generation of allophonic rules. Technical Report 0401, INRIA.

Michael R. Brent and Timothy A. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125.

Anthony C. C. Coolen, Reimer Kühn, and Peter Sollich. 2005. *Theory of Neural Information Processing Systems*. Oxford University Press.

Isabelle Dautriche. 2009. Modélisation des processus d'acquisition du langage par des méthodes statistiques. Master's thesis, INSA, Toulouse.

Brian Dillon, Ewan Dunbar, and William Idsardi. In preparation. A single stage approach to learning phonological categories: insights from inuktitut.

Charles Hockett. 1955. A manual of phonology. *International Journal of American Linguistics*, 21(4).

Rozenn Le Calvez. 2007. *Approche computationnelle de l'acquisition précoce des phonèmes*. Ph.D. thesis, UPMC, Paris.

David Marr. 1982. *Vision: a Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman.

Andrew T. Martin, Sharon Peperkamp, and Emmanuel Dupoux. Submitted. Learning phonemes with a pseudo-lexicon.

Brian W. Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta, Protein Structure*, 405(2):442–451.

Jessica Maye, Janet F. Werker, and LouAnn Gerken. 2002. Infant sensitivity to distributional information can affect phonetic discrimination. *Cognition*, 82(3):B101–B111.

Sharon Peperkamp, Rozenn Le Calvez, Jean-Pierre Nadal, and Emmanuel Dupoux. 2006. The acquisition of allophonic rules: statistical learning with linguistic constraints. *Cognition*, 101(3):B31–B41.

Jenny R. Saffran, Richard N. Aslin, and Elissa L. Newport. 1996. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928.

Tobias Sing, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. 2005. ROCR: visualizing classifier performance in R. *Bioinformatics*, 21(20):3940–3941.

Anand Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):351–372.

Turn-Taking Cues in a Human Tutoring Corpus

Heather Friedberg
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA, 15260, USA
friedberg@cs.pitt.edu

Abstract

Most spoken dialogue systems are still lacking in their ability to accurately model the complex process that is human turn-taking. This research analyzes a human-human tutoring corpus in order to identify prosodic turn-taking cues, with the hopes that they can be used by intelligent tutoring systems to predict student turn boundaries. Results show that while there was variation between subjects, three features were significant turn-yielding cues overall. In addition, a positive relationship between the number of cues present and the probability of a turn yield was demonstrated.

1 Introduction

Human conversation is a seemingly simple, everyday phenomenon that requires a complex mental process of turn-taking, in which participants manage to yield and hold the floor with little pause in-between speaking turns. Most linguists subscribe to the idea that this process is governed by a subconscious internal mechanism, that is, a set of cues or rules that steers humans toward proper turn-taking (Duncan, 1972). These cues may include lexical features such as the words used to end the turn, or prosodic features such as speaking rate, pitch, and intensity (Cutler and Pearson, 1986).

While successful turn-taking is fairly easy for humans to accomplish, it is still difficult for models to be implemented in spoken dialogue systems. Many systems use a set time-out to decide when a user is finished speaking, often resulting in unnaturally long pauses or awkward overlaps (Ward, et. al., 2005). Others detect when a user interrupts the system, known as "barge-in", though this is characteristic of failed turn-taking rather than successful conversation (Glass, 1999).

Improper turn-taking can often be a source of user discomfort and dissatisfaction with a spoken dialogue system. Little work has been done to study turn-taking in tutoring, so we hope to investigate it further while using a human-human (HH) tutoring corpus and language technologies to extract useful information about turn-taking cues. This analysis is particularly interesting in a tutoring domain because of the speculated unequal statuses of participants. The goal is to eventually develop a model for turn-taking based on this analysis which can be implemented in an existent tutoring system, ITSPOKE, an intelligent tutor for college-level Newtonian physics (Litman and Silliman, 2004). ITSPOKE currently uses a time-out to determine the end of a student turn and does not recognize student barge-in. We hypothesize that improving upon the turn-taking model this system uses will help engage students and hopefully lead to increased student learning, a standard performance measure of intelligent tutoring systems (Litman et. al., 2006).

2 Related Work

Turn-taking has been a recent focus in spoken dialogue system work, with research producing many different models and approaches. Raux and Eskenazi (2009) proposed a finite-state turn-taking

Proceedings of the ACL-HLT 2011 Student Session, pages 94–98,
Portland, OR, USA 19-24 June 2011.

model, which is used to predict end-of-turn and performed significantly better than a fixed-threshold baseline in reducing endpointing latency in a spoken dialogue system. Selfridge and Heeman (2010) took a different approach and presented a bidding model for turn-taking, in which dialogue participants compete for the turn based on the importance of what they will say next.

Of considerable inspiration to the research in this paper was Gravano and Hirschberg's (2009) analysis of their games corpus, which showed that it was possible for turn-yielding cues to be identified in an HH corpus. A similar method was used in this analysis, though it was adapted based on the tools and data that were readily available for our corpus. Since these differences may prevent direct comparison between corpora, future work will focus on making our method more analogous.

Since our work is similar to that done by Gravano and Hirschberg (2009), we hypothesize that turn-yielding cues can also be identified in our HH tutoring corpus. However, it is possible that the cues identified will be very different, due to factors specific to a tutoring environment. These include, but are not limited to, status differences between the student and tutor, engagement of the student, and the different goals of the student and tutor.

Our hypothesis is that for certain prosodic features, there will be a significant difference between places where students yield their turn (allow the tutor to speak) and places where they hold it (continue talking). This would designate these features as turn-taking cues, and would allow them to be used as features in a turn-taking model for a spoken dialogue system in the future.

3 Method

The data for this analysis is from an HH tutoring corpus recorded during the 2002-2003 school year. This is an audio corpus of 17 university students, all native Standard English speakers, working with a tutor (the same for all subjects) on physics problems (Litman et. al., 2006). Both the student and the tutor were sitting in front of separate work stations, so they could communicate only through microphones or, in the case of a student-written essay, through the shared computer environment. Any potential turn-taking cues that the tutor received from the student were very comparable to what a spoken dialogue system would have to analyze during a user interaction.

For each participant, student speech was isolated and segmented into breath groups. A breath group is defined as any segment of speech by one dialogue participant bounded by 200 ms of silence or more based on a certain threshold of intensity (Liscombe et. al., 2005). This break-down allowed for feature measurement and comparison at places that were and were not turn boundaries. Although Gravano and Hirschberg (2009) segmented their corpus by 50 ms of silence, we used 200 ms to divide the breath groups, as this data had already been calculated for another experiment done with the HH corpus (Liscombe et. al., 2005). [1]

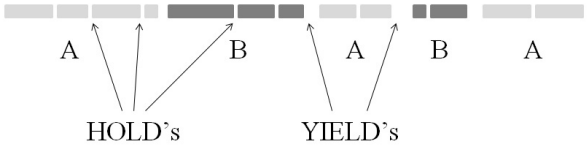

Figure 1. Conversation Segmented into Breath Groups

Each breath group was automatically labeled as one of the following: HOLD, when a breath group was immediately followed by a second breath group from the same person, YIELD, when a breath group was immediately followed by speech from the other participant, or OVERLAP, when speech from another participant started before the current one ended. Figure 1 is a diagram of a hypothetical conversation between two participants, with examples of HOLD's and YIELD's labeled. These groups were determined strictly by time and not by the actually speech being spoken. Speech acts such as backchannels, then, would be included in the YIELD group if they were spoken during clear gaps in the tutor's speech, but would be placed in the OVERLAP group if they occurred during or overlapping with tutor speech. There were 9,169 total HOLD's in the corpus and 4,773 YIELD's; these were used for comparison, while the OVERLAP's were set aside for future work.

Four prosodic features were calculated for each breath group: duration, pitch, RMS, and percent silence. Duration is the length of the breath group in seconds. Pitch is the mean fundamental frequency (f0) of the speech. RMS (the root mean

[1] Many thanks to the researchers at Columbia University for providing the breath group data for this corpus.

	N	duration	percent silence	pitch	RMS
HOLD Group Mean	993	1.07	0.34	102.24	165.27
YIELD Group Mean	480	0.78	0.39	114.87	138.89
Significance		* p = 0.018	* p < 0.001	* p < 0.001	* p < 0.001

Table 1. Individual Results for Subject 111
* denotes a significant p value

	N	duration	percent silence	pitch	RMS
HOLD Group Mean	17	1.49	0.300	140.44	418.00
YIELD Group Mean	17	0.82	0.310	147.58	354.65
Significance		* p = 0.022	p = 0.590	* p = 0.009	* p < 0.001

Table 2. Results from Paired T-Test

squared amplitude) is the energy or loudness. Percent silence was the amount of internal silence within the breath group. For pitch and RMS, the mean was taken over the length of the breath group. These features were used because they are similar to those used by Gravano and Hirschberg (2009), and are already used in the spoken dialogue system we will be using (Forbes-Riley and Litman, 2011). While only a small set of features is examined here, future work will include expanding the feature set.

Mean values for each feature for HOLD's and YIELD's were calculated and compared using the student T-test in SPSS Statistics software. Two separate tests were done, one to compare the means for each student individually, and one to compare the means across all students. $p \leq .05$ is considered significant for all statistical tests. The p-values given are the probability of obtaining the difference between groups by chance.

4 Results

4.1 Individual Cues

First, means for each feature for HOLD's and YIELD's were compared for each subject individually. These individual results indicated that while turn-taking cues could be identified, there was much variation between students. Table 1 displays the results of the analysis for one subject, student 111. For this student, all four prosodic features are turn-taking cues, as there is a significant different between the HOLD and YIELD groups for all of them. However, for all other students, this was not the case. As shown in Table 3, multiple significant cues could be identified for most students, and there was only one which appeared to have no significant turn-yielding cues.

Because there was so much individual variation, a paired T-test was used to compare the means across subjects. In this analysis, duration, pitch, and RMS were all found to be significant cues. Percent silence, however, was not. The results of this test are summarized in Table 2. A more detailed look at each of the three significant cues is done below.

Number of Significant Cues	Number of Students
0	1
1	0
2	6
3	9
4	1

Table 3. Number of Students with Significant Cues

Duration: The mean duration for HOLD's is longer than the mean duration for YIELD's. This suggests that students speak for a longer uninterrupted time when they are trying to hold their turn, and yield their turns with shorter utterances. This is the opposite of Gravano and Hirschberg's (2009) results, which found that YIELD's were longer.

Pitch: The mean pitch for YIELD's is higher than the mean pitch for HOLD's. Gravano and Hirschberg (2009), on the other hand, found that YIELD's were lower pitched than HOLD's. This difference may be accounted for by the difference in tasks. During tutoring, students are possibly

more uncertain, which may raise the mean pitch of the YIELD breath groups.

RMS: The mean RMS, or energy, for HOLD's is higher than the mean energy for YIELD's. This is consistent with student's speaking more softly, i.e., trailing off, at the end of their turn, a usual phenomenon in human speech. This is consistent with the results from the Columbia games corpus (Gravano and Hirshberg, 2009).

4.2 Combining Cues

Gravano and Hirschberg (2009) were able to show using their cues and corpus that there is a positive relationship between the number of turn-yielding cues present and the probability of a turn actually being taken. This suggests that in order to make sure that the other participant is aware whether the turn is going to continue or end, the speaker may subconsciously give them more information through multiple cues.

To see whether this relationship existed in our data, each breath group was marked with a binary value for each significant cue, representing whether the cue was present or not present within that breath group. A cue was considered present if the value for that breath group was strictly closer to the student's mean for YIELD's than HOLD's. The number of cues present for each breath group was totaled. Only the three cues found to be significant cues were used for these calculations. For each number of cues possible x (0 to 3, inclusively), the probability of the turn being taken was calculated by $p(x) = Y / T$, where Y is the number of YIELD's with x cues present, and T is the total number of breath groups with x cues present.

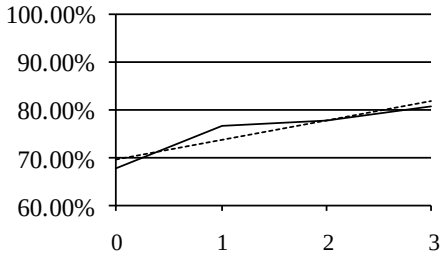

Figure 2. Cues Present v. Probability of YIELD

According to these results, a positive relationship seems to exist for these cues and this corpus. Figure 2 shows the results plotted with a fitted regression. The number of cues present and probability of a turn yield is strongly correlated ($r = .923$, $p=.038$). A regression analysis done using SPSS showed that the adjusted $r^2 = .779$ ($p = .077$).

When no turn-yielding cues are present, there is still a majority chance that the student will yield their turn; however, this is understandable due to the small number of cues being analyzed. Regardless, this gives a very preliminary support for the idea that it is possible to predict when a turn will be taken based on the number of cues present.

5 Conclusions

This paper presented preliminary work in using an HH tutoring corpus to construct a turn-taking model that can later be implemented in a spoken dialogue system. A small set of prosodic features was used to try and identify turn-taking cues by comparing their values at places where students yielded their turn to the tutor and places where they held it. Results show that turn-taking cues such as those investigated can be identified for the corpus, and may hold predictive ability for turn boundaries.

5.1 Future Work

When building on this work, there are two different directions in which we can go. While this work uncovers some interesting results in the tutoring domain, there are some shortcomings in the method that may make it difficult to effectively evaluate the results. As the breath group is different from the segment used in Gravano and Hirschberg's (2009) experiment, and the set of prosodic features is smaller, *direct* comparison becomes quite difficult. The differences between the two methods provide enough doubt for the results to truly be interpreted as contradictory. Thus the first line of future inquiry is to redo this method using a smaller silence boundary (50 ms) and different set of prosodic features so that it is truly comparable to Gravano and Hirschberg's (2009) work with the game corpus. This could yield interesting discoveries in the differences between the two corpora, shedding light on phenomena that are particular to tutoring scenarios.

On the other hand, other researchers have used different segments; for example, Clemens and Diekhaus (2009) divide their corpus by "topic units" that are grammatically and semantically complete. In addition, Litman et. al. (2009) were able to use

word-level units to calculate prosody and classify turn-level uncertainty. Perhaps direct comparison is not entirely necessary, and instead this work should be considered an isolated look at an HH corpus that provides insight into turn-taking, specifically in tutoring and other domains with unequal power levels. Future work in this direction would include growing the set of features by adding more prosodic ones and introducing lexical ones such as bi-grams and uni-grams. Already, work has been done to investigate the features used in the INTERSPEECH 2009 Emotion Challenge using openSMILE (Eyben et. al., 2009). When a large feature bank has been developed, significant cues will be used in conjunction with machine learning techniques to build a model for turn-taking which can be implemented in a spoken dialogue tutoring system. The goal would be to learn more about human turn-taking while seeing if better turn-taking by a computer tutor ultimately leads to increased student learning in an intelligent tutoring system.

Acknowledgments

This work was supported by the NSF (#0631930). I would like to thank Diane Litman, my advisor, Scott Silliman, for software assistance, Joanna Drummond, for many helpful comments on this paper, and the ITSPOKE research group for their feedback on my work.

References

Caroline Clemens and Christoph Diekhaus. 2009. Prosodic turn-yielding Cues with and without optical Feedback. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics.

Anne Cutler and Mark Pearson. 1986. On the analysis of prosodic turn-taking cues. In C. Johns-Lewis, Ed., *Intonation in Discourse,* pp. 139-156. College-Hill.

Starkey Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology,* 24(2):283-292.

Florian Eyben, Martin Wöllmer, Björn Schuller. 2010. openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor. Proc. ACM Multimedia (MM), ACM, Florence, Italy. pp. 1459-1462.

James R. Glass. 1999. Challenges for spoken dialogue systems. In *Proceedings of the 1999 IEEE ASRU Workshop.*

Agustín Gravano and Julia Hirschberg. 2009. Turn-yielding cues in task-oriented dialogue. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 253--261. Association for Computational Linguistics.

Jackson Liscombe, Julia Hirschberg, and Jennifer J. Venditti. 2005. Detecting certainness in spoken tutorial dialogues. In *Interspeech.*

Diane J. Litman, Carolyn P. Rose, Kate Forbes-Riley, Kurt VanLehn, Dumisizwe Bhembe, and Scott Silliman. 2006. Spoken Versus Typed Human and Computer Dialogue Tutoring. In *International Journal of Artificial Intelligence in Education*, 26: 145-170.

Diane Litman, Mihai Rotaru, and Greg Nicholas. 2009. Classifying Turn-Level Uncertainty Using Word-Level Prosody. *Proceedings Interspeech*, Brighton, UK, September.

Kate Forbes-Riley and Diane Litman. 2011. Benefits and Challenges of Real-Time Uncertainty Detection and Adaptation in a Spoken Dialogue Computer Tutor. *Speech Communication*, in press.

Diane J. Litman and Scott Silliman. 2004. Itspoke: An intelligent tutoring spoken dialogue system. In *HLT/NAACL*.

Antoine Raux and Maxine Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proc. NAACL/HLT 2009*, Boulder, CO, USA.

Ethan O. Selfridge and Peter A. Heeman. 2010. Importance-Driven Turn-Bidding for spoken dialogue systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (ACL '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 177-185.

Nigel Ward, Anais Rivera, Karen Ward, and David G. Novick. 2005. Root causes of lost time and user stress in a simple dialog system. In *Proceedings of Interspeech.*

Predicting Clicks in a Vocabulary Learning System

Aaron Michelony
Baskin School of Engineering
University of California, Santa Cruz
1156 High Street
Santa Cruz, CA 95060
amichelo@soe.ucsc.edu

Abstract

We consider the problem of predicting which words a student will click in a vocabulary learning system. Often a language learner will find value in the ability to look up the meaning of an unknown word while reading an electronic document by clicking the word. Highlighting words likely to be unknown to a reader is attractive due to drawing his or her attention to it and indicating that information is available. However, this option is usually done manually in vocabulary systems and online encyclopedias such as Wikipedia. Furthurmore, it is never on a per-user basis. This paper presents an automated way of highlighting words likely to be unknown to the specific user. We present related work in search engine ranking, a description of the study used to collect click data, the experiment we performed using the random forest machine learning algorithm and finish with a discussion of future work.

1 Introduction

When reading an article one occasionally encounters an unknown word for which one would like the definition. For students learning or mastering a language, this can occur frequently. Using a computerized learning system, it is possible to highlight words with which one would expect students to struggle. The highlighting both draws attention to the word and indicates that information about it is available.

There are many applications of automatically highlighting unknown words. The first is, obviously, educational applications. Another application is foreign language acquisition. Traditionally learners of foreign languages have had to look up unknown words in a dictionary. For reading on the computer, unknown words are generally entered into an online dictionary, which can be time-consuming. The automated highlighting of words could also be applied in an online encyclopedia, such as Wikipedia. The proliferation of handheld computer devices for reading is another potential application, as some of these user interfaces may cause difficulty in the copying and pasting of a word into a dictionary. Given a finite amount of resources available to improve definitions for certain words, knowing which words are likely to be clicked will help. This can be used for caching.

In this paper, we explore applying machine learning algorithms to classifying clicks in a vocabulary learning system. The primary contribution of this work is to provide a list of features for machine learning algorithms and their correlation with clicks. We analyze how the different features correlate with different aspects of the vocabulary learning process.

2 Related Work

The previous work done in this area has mainly been in the area of predicting clicks for web search ranking. For search engine results, there have been several factors identified for why people click on certain results over others. One of the most important is position bias, which says that the presentation order affects the probability of a user clicking on a result. This is considered a "fundamental problem in click data" (Craswell et al., 2008), and eye-

Proceedings of the ACL-HLT 2011 Student Session, pages 99–104,
Portland, OR, USA 19-24 June 2011.

tracking experiments (Joachims et al., 2005) have shown that click probability decays faster than examination probability.

There have been four hypotheses for how to model position bias:

- Baseline Hypothesis: There is no position bias. This may be useful for some applications but it does not fit with the data for how users click the top results.
- Mixture Hypothesis: Users click based on relevance or at random.
- Examination Hypothesis: Each result has a probability of being examined based on its position and will be clicked if it is both examined and relevant.
- Cascade Model: Users view search results from top to bottom and click on a result with a certain probability.

The cascade model has been shown to closely model the top-ranked results and the baseline model closely matches how users click at lower-ranked results (Craswell et al., 2008).

There has also been work done in predicting document keywords (Doğan and Lu, 2010). Their approach is similar in that they use machine learning to recognize words that are important to a document. Our goals are complimentary, in that they are trying to predict words that a user would use to search for a document and we are trying to predict words in a document that a user would want more information about. We revisit the comparison later in our discussion.

3 Data Description

To obtain click data, a study was conducted involving middle-school students, of which 157 were in the 7th grade and 17 were in the 8th grade. 90 students spoke Spanish as their primary language, 75 spoke English as their primary language, 8 spoke other languages and 1 was unknown. There were six documents for which we obtained click data. Each document was either about science or was a fable. The science documents contained more advanced vocabulary whereas the fables were primarily written for English language learners. In the study, the students took a vocabulary test, used the vocabulary system and then took another vocabulary test with the same words. The highlighted words were chosen by a computer program using latent semantic analysis (Deerwester et al., 1990) and those results were then manually edited by educators. The words were highlighted identically for each student. Importantly, only nouns were highlighted and only nouns were in the vocabulary test. When the student clicked on a highlighted word, they were shown definitions for the word along with four images showing the word in context. For example, if a student clicked on the word “crane” which had the word “flying” next to it, one of the images the student would see would be of a flying crane. From Figure 1 we see that there is a relation between the total number of words in a document and the number of clicks students made.

Number	Genre	Words	Students
1	Science	2935	60
2	Science	2084	138
3	Fable	667	23
4	Fable	513	22
5	Fable	397	16
6	Fable	105	5

Table 1. Document Information

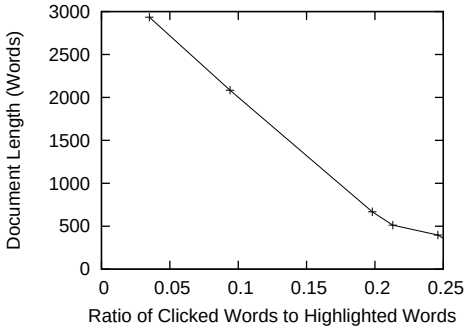

Figure 1. Document Length Affects Clicks

It should be noted that there is a large class imbalance in the data. For every click in document four, there are about 30 non-clicks. The situation is even more imbalanced for the science documents. For the second science document there are 100 non-clicks for every click and for the first science document there are nearly 300 non-clicks for every click.

There was also no correlation seen between a word being on a quiz and being clicked. This indicates that the students may not have used the system as seriously as possible and introduced noise into the click data. This is further evidenced by the quizzes, which show that only about 10% of the quiz words that students got wrong on the first test were actually learned. However, we will show that we are able to predict clicks regardless.

Figure 2, 3 and 4 show the relationship between the mean age of acquisition of the words clicked on, STAR language scores and the number of clicks for document 2. A second-degree polynomial was fit to the data for each figure. Students with STAR language scores above 300 are considered to have basic ability, above 350 are proficient and above 400 are advanced. Age of acquisition scores are abstract and a score of 300 means a word was acquired at 4-6, 400 is 6-8 and 500 is 8-10 (Cortese and Fugett, 2004).

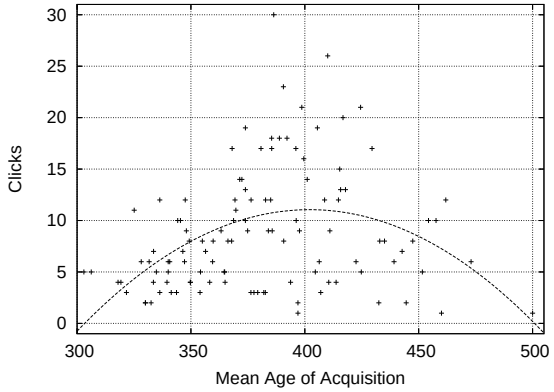

Figure 2. Age of Acquisition vs Clicks

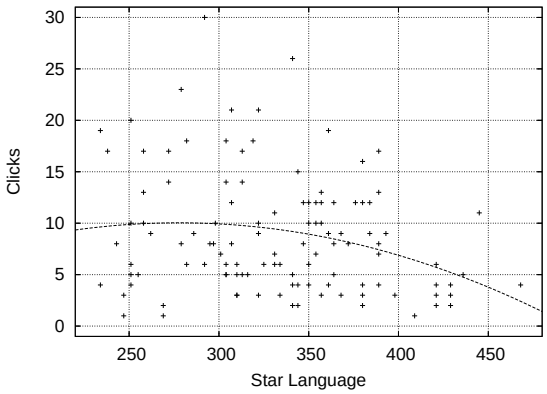

Figure 3. STAR Language vs Clicks

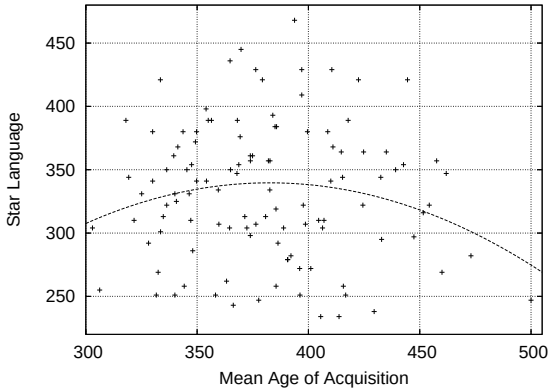

Figure 4. Age of Acquisition vs STAR Language

4 Machine Learning Method

The goal of our study is to predict student clicks in a vocabulary learning system. We used the random forest machine learning method, due to its success in the Yahoo! Learning to Rank Challenge (Chapelle and Chang, 2011). This algorithm was tested using the Weka (Hall et al., 2009) machine learning software with the default settings.

Random forest is an algorithm that classifies data by decision trees voting on a classification (Breiman, 2001). The forest chooses the class with the most votes. Each tree in the forest is trained by first sampling a subset of the data, chosen randomly with replacement, and then removing a large number of features. The number of samples chosen is the same number as in the original dataset, which usually results in about one-third of the original dataset left out of the training set. The tree is unpruned. Random forest has the advantage that it does not overfit the data.

To implement this algorithm on our click data, we constructed feature vectors consisting of both student features and word features. Each word is either clicked or not clicked, so we were able to use a binary classifier.

5 Evaluation

5.1 Features

To run our machine learning algorithms, we needed features for them. The features used are of two types: student features and word features. The student features we used in our experiment were the STAR (Standardized Testing and Reporting, a California standardized test) language score and the CELDT (California English Language Development Test) overall score, which correlated highly with each other. There was a correlation of about -0.1 between the STAR language score and total clicks across all the documents. Also available were the STAR math score, CELDT reading, writing, speaking and listening scores, grade level and primary language. These did not improve results and were not included in the experiment.

We used and tested many word features, which were discovered to be more important than the student features. First, we used the part-of-speech as a feature which was useful since only nouns were highlighted in the study. The part-of-speech tagger we used was the Stanford Log-linear Part-of-Speech Tagger (Toutanova et al., 2003). Second, various psycholinguistic variables were obtained from five studies (Wilson, 1988; Bird et al., 2001; Cortese and Fugett, 2004; Stadthagen-Gonzalez and Davis, 2006; Cortese and Khanna, 2008). The most useful was age of acquisition, which refers to "the age at which a word was learnt and has been proposed as a significant contributor to language and memory processes" (Stadthagen-Gonzalez and Davis, 2006). This was useful because it was available for the majority of words and is a good proxy for the difficulty of a word. Also useful was imageability, which is "the ease with which the word gives rise to a sensory mental image" (Bird et al., 2001). For example, these words are listed in decreasing order of imageability: beach, vest, dirt, plea, equanimity. Third, we obtained the Google unigram frequencies which were also a proxy for the difficulty of a word. Fourth, we calculated click percentages for words, students and words, words in a document and specific words in a document. While these features correlated very highly with clicks, we did not include these in our experiment. We instead would like to focus on words for which we do not have click data. Fifth, the word position, which indicates the position of the word in the document, was useful because position bias was seen in our data. Also important was the word instance, e.g. whether the word is the first, second, third, etc. time appearing in the document. After seeing a word three or four times, the clicks for that word dropped off dramatically.

There were also some other features that seemed interesting but ultimately proved not useful. We gathered etymological data, such as the language of origin and the date the word entered the English language; however these features did not help. We were also able to categorize the words using WordNet (Fellbaum, 1998), which can determine, for example, that a boat is an artifact and a lion is an animal. We tested for the categories of abstraction, artifact, living thing and animal but found no correlation between clicks and these categories.

5.2 Missing Values

Many features were not available for every word in the evaluation, such as age of acquisition. We could guess a value from available data, called imputation, or create separate models for each unique pattern of missing features, called reduced-feature models. We decided to create reduced feature models due to them being reported to consistently outperform imputation (Saar-Tsechansky and Provost, 2007).

5.3 Experimental Set-up

We ran our evaluation on document four, which had click data for 22 students. We chose this document because it had the highest correlation between a word being a quiz word and clicked, at 0.06, and the correlation between the age of acquisition of a word and that word being a quiz word is high, at 0.58.

The algorithms were run with the following features: STAR language score, CELDT overall score, word position, word instance, document number, age of acquisition, imageability, Google frequency, stopword, and part-of-speech. We did not include the science text data as training data. The training data for a student consisted of his or her click data for the other fables and all the other students' click data for all the fables.

5.4 Results

From Figure 2 we see the performance of random forest. We obtained similar performance with the other documents except document one. We also note that we also used a bayesian network and multi-boosting in Weka and obtained similar performance to random forest.

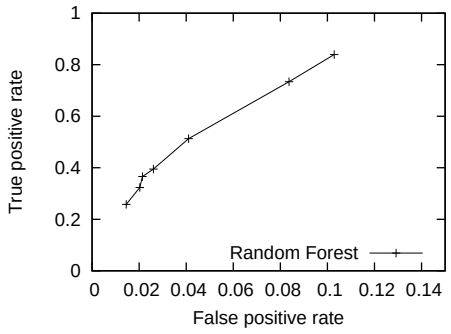

Figure 5. ROC Curve of Results

6 Discussion

There are several important issues to consider when interpreting these results. First, we are trying to maximize clicks when we should be trying to maximize learning. In the future we would like to identify which clicks are more important than others and incorporate that into our model. Second, across all documents of the study there was no correlation between a word being on the quiz and being clicked. We would like to obtain click data from users actively trying to learn and see how the results would be affected and we speculate that the position bias effect may be reduced in this case. Third, this study involved students who were using the system for the first time. How these results translate to long-term use of the program is unknown.

The science texts are a challenge for the classifiers for several reasons. First, due to the relationship between a document's length and the number of clicks, there are relatively few words clicked. Second, in the study most of the more difficult words were not highlighted. This actually produced a slight negative correlation between age of acquisition and whether the word is a quiz word or not, whereas for the fable documents there is a strong positive correlation between these two variables. It raises the question of how appropriate it is to include click data from a document with only one click out of 100 or 300 non-clicks into the training set for a document with one click out of 30 non-clicks. When the science documents were included in the training set for the fables, there was no difference in performance.

The correlation between the word position and clicks is about -0.1. This shows that position bias affects vocabulary systems as well as search engines and finding a good model to describe this is future work. The cascade model seems most appropriate, however the students tended to click in a non-linear order. It remains to be seen whether this non-linearity holds for other populations of users.

Previous work by Doğan and Lu in predicting click-words (Doğan and Lu, 2010) built a learning system to predict click-words for documents in the field of bioinformatics. They claim that "Our results show that a word's semantic type, location, POS, neighboring words and phrase information together could best determine if a word will be a click-word." They did report that if a word was in the title or abstract it was more likely to be a click-word, which is similar to our finding that a word at the beginning of the document is more likely to be clicked. However, it is not clear whether there is one underlying cause for both of these. Certain features such as neighboring words do not seem applicable to our usage in general, although it is something to be aware of for specialized domains. Their use of semantic types was interesting, though using WordNet we did not find any preference for certain classes of nouns being clicked over others.

Acknowledgements

I would like to thank Yi Zhang for mentoring and providing ideas. I would also like to thank Judith Scott, Kelly Stack, James Snook and other members of the TecWave project. I would also like to think the anonymous reviewers for their helpful comments. Part of this research is funded by National Science Foundation IIS-0713111 and the Institute of Education Science. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the author, and do not necessarily reflect those of the sponsors.

References

Helen Bird, Sue Franklin, and David Howard. 2001. *Age of Acquisition and Imageability Ratings for a Large Set of Words, Including Verbs and Function Words*. Behavior Research Methods, Instruments, & Computers, 33:73-79.

Leo Breiman. 2001. *Random Forests*. Machine Learning 45(1):5-32

Olivier Chapelle and Yi Chang. 2011. *Yahoo! Learning to Rank Challenge Overview*. JMLR: Workshop and Conference Proceedings 14 1-24.

Michael J. Cortese and April Fugett. 2004. *Imageability Ratings for 3,000 Monosyllabic Words*. Behavior Research Methods, Instruments, and Computers, 36:384-387.

Michael J. Cortese and Maya M. Khana. 2008. *Age of Acquisition Ratings for 3,000 Monosyllabic Words*. Behavior Research Methods, 40:791-794.

Nick Craswell, Onno Zoeter, Michael Taylor, Bill Ramsey. 2008. *An Experimental Comparison of Click Position-Bias Models*. First ACM International Conference on Web Search and Data Mining WSDM 2008.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman. 1990. *Indexing by Latent Semantic Analysis*. Journal of the American Society for Information Science, 41(6):391-407.

Rezarta I. Doğan and Zhiyong Lu. 2010. *Click-words: Learning to Predict Document Keywords from a User Perspective*. Bioinformatics, 26, 2767-2775.

Christine Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Yoav Freund and Robert E. Shapire. 1995. *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting*. Journal of Computer and System Sciences, 55:119-139.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten 2009. *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, Volume 11, Issue 1.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Geri Gay. 2005. *Accurately Interpreting Clickthrough Data as Implicit Feedback*. Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR), 2005.

Maytal Saar-Tsechansky and Foster Provost. 2007. *Handling Missing Values when Applying Classication Models*. The Journal of Machine Learning Research, 8:1625-1657.

Hans Stadthagen-Gonzalez and Colin J. Davis. 2006. *The Bristol Norms for Age of Acquisition, Imageability and Familiarity*. Behavior Research Methods, 38:598-605.

Kristina Toutanova, Dan Klein, Christopher Manning, Yoram Singer. 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. Proceedings of HLT-NAACL 2003, 252-259.

Michael D. Wilson. 1988. *The MRC Psycholinguistic Database: Machine Readable Dictionary, Version 2*. Behavioural Research Methods, Instruments and Computers, 20(1):6-11.

Exploiting Morphology in Turkish Named Entity Recognition System

Reyyan Yeniterzi *
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
reyyan@cs.cmu.edu

Abstract

Turkish is an agglutinative language with complex morphological structures, therefore using only word forms is not enough for many computational tasks. In this paper we analyze the effect of morphology in a Named Entity Recognition system for Turkish. We start with the standard word-level representation and incrementally explore the effect of capturing syntactic and contextual properties of tokens. Furthermore, we also explore a new representation in which roots and morphological features are represented as separate tokens instead of representing only words as tokens. Using syntactic and contextual properties with the new representation provide an 7.6% relative improvement over the baseline.

1 Introduction

One of the main tasks of information extraction is the Named Entity Recognition (NER) which aims to locate and classify the named entities of an unstructured text. State-of-the-art NER systems have been produced for several languages, but despite all these recent improvements, developing a NER system for Turkish is still a challenging task due to the structure of the language.

Turkish is a morphologically complex language with very productive inflectional and derivational processes. Many local and non-local syntactic structures are represented as morphemes which at the end produces Turkish words with complex morphological structures. For instance, the following English phrase *"if we are going to be able to make [something] acquire flavor"* which contains the necessary function words to represent the meaning can be translated into Turkish with only one token *"tatlandırabileceksek"* which is produced from the root "tat" (flavor) with additional morphemes +lan (acquire), +dır (to make), +abil (to be able), +ecek (are going), +se (if) and +k (we).

This productive nature of the Turkish results in production of thousands of words from a given root, which cause data sparseness problems in model training. In order to prevent this behavior in our NER system, we propose several features which capture the meaning and syntactic properties of the token in addition to the contextual properties. We also propose using a sequence of morphemes representation which uses roots and morphological features as tokens instead of words.

The rest of this paper is organized as follows: Section 2 summarizes some previous related works, Section 3 describes our approach, Section 4 details the data sets used in the paper, Section 5 reports the experiments and results and Section 6 concludes with possible future work.

2 Related Work

The first paper (Cucerzan and Yarowski, 1999) on Turkish NER describes a language independent bootstrapping algorithm that learns from word internal and contextual information of entities. Turkish was one of the five languages the authors experimented with. In another work (Tur et al., 2003),

*The author is also affiliated with iLab and the Center for the Future of Work of Heinz College, Carnegie Mellon University

Proceedings of the ACL-HLT 2011 Student Session, pages 105–110,
Portland, OR, USA 19-24 June 2011.

the authors followed a statistical approach (HMMs) for NER task together with some other Information Extraction related tasks. In order to deal with the agglutinative structure of the Turkish, the authors worked with the root-morpheme level of the word instead of the surface form. A recent work (Kücük and Yazici, 2009) presents the first rule-based NER system for Turkish. The authors used several information sources such as dictionaries, list of well known entities and context patterns.

Our work is different from these previous works in terms of the approach. In this paper, we present the first CRF-based NER system for Turkish. Furthermore, all these systems used word-level tokenization but in this paper we present a new tokenization method which represents each root and morphological feature as separate tokens.

3 Approach

In this work, we used two tokenization methods. Initially we started with the sequence of words representation which will be referred as word-level model. We also introduced morpheme-level model in which morphological features are represented as states. We used several features which were created from deep and shallow analysis of the words. During our experiments we used Conditional Random Fields (CRF) which provides advantages over HMMs and enables the use of any number of features.

3.1 Word-Level Model

Word-level tokenization is very commonly used in NER systems. In this model, each word is represented with one state. Since CRF can use any number of features to infer the hidden state, we develop several feature sets which allow us to represent more about the word.

3.1.1 Lexical Model

In this model, only the word tokens are used in their surface form. This model is effective for many languages which do not have complex morphological structures. However for morphologically rich languages, further analysis of words is required in order to prevent data sparseness problems and produce more accurate NER systems.

3.1.2 Root Feature

An analysis (Hakkani-Tür, 2000) on English and Turkish news articles with around 10 million words showed that on the average 5 different Turkish word forms are produced from the same root. In order to decrease this high variation of words we use the root forms of the words as an additional feature.

3.1.3 Part-of-Speech and Proper-Noun Features

Named entities are mostly noun phrases, such as first name and last name or organization name and the type of organization. This property has been used widely in NER systems as a hint to determine the possible named entities.

Part-of-Speech tags of the words depend highly on the language and the available Part-of-Speech tagger. Taggers may distinguish the proper nouns with or without their types. We used a Turkish morphological analyzer (Oflazer, 1994) which analyzes words into roots and morphological features. An example to the output of the analyzer is given in Table 1. The part-of-speech tag of each word is also reported by the tool [1]. We use these tags as additional features and call them part-of-speech (POS) features.

The morphological analyzer has a proper name database, which is used to tag Turkish person, location and organization names as proper nouns. An example name entity with this *+Prop* tag is given in Table 1. Although, the use of this tag is limited to the given database and not all named entities are tagged with it, we use it as a feature to distinguish named entities. This feature is referred as proper-noun (Prop) feature.

3.1.4 Case Feature

As the last feature, we use the orthographic case information of the words. The initial letter of most named entities is in upper case, which makes case feature a very common feature in NER tasks. We also use this feature and mark each token as *UC* or *LC* depending on the initial letter of it. We don't do

[1] The meanings of various Part-of-Speech tags are as follows: **+A3pl** - 3rd person plural; **+P3sg** - 3rd person singular possessive; **+Gen** - Genitive case; **+Prop** - Proper Noun; **+A3sg** - 3rd person singular; **+Pnon** - No possesive agreement; **+Nom** - Nominative case.

Table 1: Examples to the output of the Turkish morphological analyzer

WORD	+	**ROOT**	+	**POS**	+	**MORPHEMES**
beyinlerinin (*of their brains*)	+	beyin	+	Noun	+	A3pl+P3sg+Gen
Amerika (*America*)	+	Amerika	+	Noun	+	Prop+A3sg+Pnon+Nom

anything special for the first words in sentences.

An example phase in word-level model is given in Table 2 [2]. In the figure each row represents a state. The first column is the lexical form of the word and the rest of the columns are the features and the tag is in the last column.

3.2 Morpheme-Level Model

Using Part-of-Speech tags as features introduces some syntactic properties of the word to the model, but still there is missing information of other morphological tags such as number/person agreements, possessive agreements or cases. In order to see the effect of these morphological tags in NER, we propose a morpheme-level tokenization method which represents a word in several states; one state for a root and one state for each morphological feature.

In a setting like this, the model has to be restricted from assigning different labels to different parts of the word. In order to do this, we use an additional feature called root-morph feature. The root-morph is a feature which is assigned the value *"root"* for states containing a root and the value *"morph"* for states containing a morpheme. Since there are no prefixes in Turkish, a model trained with this feature will give zero probability (or close to zero probability if there is any smoothing) for assigning any B-* (Begin any NE) tag to a morph state. Similarly, transition from a state with B-* or I-* (Inside any NE) tag to a morph state with O (Other) tag will get zero probability from the model.

In morpheme-level model, we use the following features:

- the actual root of the word for root and morphemes of the token
- the Part-of-speech tag of the word for the root part and the morphological tag for the morphemes
- the root-morph feature which assigns *"root"* to the roots and *"morph"* to the morphemes
- the proper-noun feature
- the case feature

An example phrase in root-morpheme-based chunking is given in Table 3. In the figure each row represents a state and each word is represented with several states. The first row of each word contains the root, POS tag and *Root* value for the root-morph feature. The rest of the rows of the same word contains the morphemes and *Morph* value for the root-morph feature.

4 Data Set

We used training set of the newspaper articles data set that has been used in (Tur et al., 2003). Since we do not have the test set they have used in their paper, we had to come up with our own test set. We used only 90% of the train data for training and left the remaining for testing.

Three types of named entities; *person*, *organization* and *location*, were tagged in this dataset. If the word is not a proper name, then it is tagged with *other*. The number of words and named entities for each NE type from train and tests sets are given in Table 4.

Table 4: The number of words and named entities in train and test set

	#WORDS	#PER.	#ORG.	#LOC.
TRAIN	445,498	21,701	14,510	12,138
TEST	47,344	2,400	1,595	1,402

5 Experiments and Results

Before using our data in the experiments we applied the Turkish morphological analyzer tool (Oflazer, 1994) and then used Morphological disambiguator (Sak et al., 2008) in order to choose the correct morphological analysis of the word depending on the

[2] One can see that *Ilias* which is Person NE is not tagged as Prop (Proper Noun) in the example, mainly because it is missing in the proper noun database of the morphological analyzer.

Table 2: An example phrase in word-level model with all features

LEXICAL	ROOT	POS	PROP	CASE	TAG
Ayvalık	Ayvalık	Noun	Prop	UC	B-LOCATION
doğumlu	doğum (*birth*)	Noun	NotProp	LC	O
yazar	yazar (*author*)	Noun	NotProp	LC	O
Ilias	ilias	Noun	NotProp	UC	B-PERSON

Table 3: An example phrase in morpheme-level model with all features

ROOT	POS	ROOT-MORPH	PROP	CASE	TAG
Ayvalık	Noun	Root	Prop	UC	B-LOCATION
Ayvalık	Prop	Morph	Prop	UC	I-LOCATION
Ayvalık	A3sg	Morph	Prop	UC	I-LOCATION
Ayvalık	Pnon	Morph	Prop	UC	I-LOCATION
Ayvalık	Nom	Morph	Prop	UC	I-LOCATION
doğum	Noun	Root	NotProp	LC	O
doğum	Adj	Morph	NotProp	LC	O
doğum	With	Morph	NotProp	LC	O
yazar	Noun	Root	NotProp	LC	O
yazar	A3sg	Morph	NotProp	LC	O
yazar	Pnon	Morph	NotProp	LC	O
yazar	Nom	Morph	NotProp	LC	O
Ilias	Noun	Root	NotProp	UC	B-PERSON
Ilias	A3sg	Morph	NotProp	UC	I-PERSON
Ilias	Pnon	Morph	NotProp	UC	I-PERSON
Ilias	Nom	Morph	NotProp	UC	I-PERSON

context. In experiments, we used CRF++ [3], which is an open source CRF sequence labeling toolkit and we used the conlleval [4] evaluation script to report F-measure, precision and recall values.

5.1 Word-level Model

In order to see the effects of the features individually, we inserted them to the model one by one iteratively and applied the model to the test set. The F-measures of these models are given in Table 5. We can observe that each feature is improving the performance of the system. Overall the F-measure was increased by 6 points when all the features are used.

5.2 Morpheme-level Model

In order to make a fair comparison between the word-level and morpheme-level models, we used all the features in both models. The results of these experiments are given in Table 6. According to the table, morpheme-level model achieved better results than word-level model in person and location entities. Even though word-level model got better F-Measure score in organization entity, morpheme-level is much better than word-level model in terms of recall.

Using morpheme-level tokenization to introduce morphological information to the model did not hurt the system, but it also did not produce a significant improvement. There may be several reasons for this. One can be that morphological information is not helpful in NER tasks. Morphemes in Turkish words are giving the necessary syntactic meaning to the word which may not be useful in named entity finding. Another reason for not seeing a significant change with morpheme usage can be our representation. Dividing the word into root and morphemes and using them as separate tokens may not be the best way of using morphemes in the model. Other ways of representing morphemes in the model may produce more effective results.

As mentioned in Section 4, we do not have the same test set that has been used in Tur et al. (Tur et al., 2003). Even though it is impossible to make a fair comparison between these two systems, it would

[3] CRF++: Yet Another CRF toolkit

[4] www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt

Table 5: F-measure Results of Word-level Model

	PERSON	ORGANIZATION	LOCATION	OVERALL
LEXICAL MODEL (LM)	80.88	77.05	88.40	82.60
LM + ROOT	83.32	80.00	90.30	84.96
LM + ROOT + POS	84.91	81.63	90.18	85.98
LM + ROOT + POS + PROP	86.82	82.66	90.52	87.18
LM + ROOT + POS + PROP + CASE	**88.58**	**84.71**	**91.47**	**88.71**

Table 6: Results of Morpheme-Level (Morp) and Word-Level Models (Word)

	PRECISION		RECALL		F-MEASURE	
	MORP	WORD	MORP	WORD	MORP	WORD
PERSON	**91.87%**	91.41%	**86.92%**	85.92%	**89.32**	88.58
ORGANIZATION	85.23%	**91.00%**	**81.84%**	79.23%	83.50	**84.71**
LOCATION	**94.15%**	92.83%	**90.23%**	90.14%	**92.15**	91.47
OVERALL	91.12%	**91.81%**	**86.87%**	85.81%	**88.94**	88.71

Table 7: F-measure Comparison of two systems

	OURS	(TUR ET AL., 2003)
BASELINE MODEL	82.60	86.01
BEST MODEL	88.94	91.56
IMPROVEMENT	**7.6%**	**6.4%**

be good to note how these systems performed with respect to their baselines which is lexical model in both. As it can be seen from Table 7, both models improved upon their baselines significantly.

6 Conclusion and Future Work

In this paper, we explored the effects of using features like root, POS tag, proper noun and case to the performance of NER task. All these features seem to improve the system significantly. We also explored a new way of including morphological information of words to the system by using several tokens for a word. This method produced compatible results to the regular word-level tokenization but did not produce a significant improvement.

As future work we are going to explore other ways of representing morphemes in the model. Here we represented morphemes as separate states, but including them as features together with the root state may produce better models. Another approach we will also focus is dividing words into characters and applying character-level models (Klein et al., 2003).

Acknowledgments

The author would like to thank William W. Cohen, Kemal Oflazer, Gökhan Tur and Behrang Mohit for their valuable feedback and helpful discussions. The author also thank Kemal Oflazer for providing the data set and the morphological analyzer. This publication was made possible by the generous support of the iLab and the Center for the Future of Work. The statements made herein are solely the responsibility of the author.

References

Silviu Cucerzan and David Yarowski. 1999. Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of the Joint SIGDAT Conference on EMNLP and VLC*, pages 90–99.

Dilek Z. Hakkani-Tür. 2000. *Statistical Language Modelling for Turkish*. Ph.D. thesis, Department of Computer Engineering, Bilkent University.

Dan Klein, Joseph Smarr, Huy Nguyen, and Christopher D. Manning. 2003. Named entity recognition with character-level models. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, pages 180–183.

Dilek Kücük and Adnan Yazici. 2009. Named entity recognition experiments on Turkish texts. In *Proceedings of the 8th International Conference on Flexible Query Answering Systems*, FQAS '09, pages 524–535, Berlin, Heidelberg. Springer-Verlag.

Kemal Oflazer. 1994. Two-level description of Turk-

ish morphology. *Literary and Linguistic Computing*, 9(2):137–148.

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *Advances in Natural Language Processing*, volume 5221 of *Lecture Notes in Computer Science*, pages 417–427.

Gökhan Tur, Dilek Z. Hakkani-Tür, and Kemal Of lazer. 2003. A statistical information extraction system for Turkish. In *Natural Language Engineering*, pages 181–210.

Social Network Extraction from Texts: A Thesis Proposal

Apoorv Agarwal
Department of Computer Science
Columbia University
apoorv@cs.columbia.edu

Abstract

In my thesis, I propose to build a system that would enable extraction of social interactions from texts. To date I have defined a comprehensive set of social events and built a preliminary system that extracts social events from news articles. I plan to improve the performance of my current system by incorporating semantic information. Using domain adaptation techniques, I propose to apply my system to a wide range of genres. By extracting linguistic constructs relevant to social interactions, I will be able to empirically analyze different kinds of linguistic constructs that people use to express social interactions. Lastly, I will attempt to make convolution kernels more scalable and interpretable.

1 Introduction

Language is the primary tool that people use for establishing, maintaining and expressing social relations. This makes language the real carrier of social networks. The overall goal of my thesis is to build a system that automatically extracts a social network from raw texts such as literary texts, emails, blog comments and news articles. I take a "social network" to be a network consisting of individual human beings and groups of human beings who are connected to each other through various relationships by the virtue of participating in *social events*. I define social events to be events that occur between people where at least one person is aware of the other and of the event taking place. For example, in the sentence *John talks to Mary*, entities *John* and *Mary* are aware of each other and of the talking event. In the sentence *John thinks Mary is great*, only *John* is aware of *Mary* and the event is the thinking event. My thesis will introduce a novel way of constructing networks by analyzing text to capture such interactions or events.

Motivation: Typically researchers construct a social network from various forms of electronic interaction records like self-declared friendship links, sender-receiver email links and phone logs etc. They ignore a vastly rich network present in the content of such sources. Secondly, many rich sources of social networks remain untouched simply because there is no meta-data associated with them (literary texts, new stories, historical texts). By providing a methodology for analyzing language to extract interaction links between people, my work will overcome both these limitations. Moreover, by empirically analyzing large corpora of text from different genres, my work will aid in formulating a comprehensive linguistic theory about the types of linguistic constructs people often use to interact and express their social interactions with others. In the following paragraphs I will explicate these impacts.

Impact on current SNA applications: Some of the current social network analysis (SNA) applications that utilize interaction meta-data to construct the underlying social network are discussed by Domingos and Richardson (2003), Kempe et al. (2003), He et al. (2006), Rowe et al. (2007), Lindamood et al. (2009), Zheleva and Getoor (2009). But meta-data captures only part of all the interactions in which people participate. There is a vastly rich network present in text such as the content of emails, comment threads on online social networks, transcribed phone calls. My work will enrich the

Proceedings of the ACL-HLT 2011 Student Session, pages 111–116,
Portland, OR, USA 19-24 June 2011.

social network that SNA community currently uses by complementing it with the finer interaction linkages present in text. For example, Rowe et al. (2007) use the sender-receiver email links to connect people in the Enron email corpus. Using this network, they predict the organizational hierarchy of the Enron Corporation. Their social network analysis for calculating centrality measure of people does not take into account interactions that people talk about in the content of emails. Such linkages are relevant to the task for two reasons. First, people talk about their interactions with other people in the content of emails. By ignoring these interaction linkages, the underlying communication network used by Rowe et al. (2007) to calculate various features is incomplete. Second, sender-receiver email links only represent "who talks to whom". They do not represent "who talks about whom to whom." This later information seems to be crucial to the task presumably because people at the lower organizational hierarchy are more likely to talk about people higher in the hierarchy. My work will enable extraction of these missing linkages and hence offers the potential to improve the performance of currently used SNA algorithms. By capturing alternate forms of communications, my system will also overcome a known limitation of the Enron email corpus that a significant number of emails were lost at the time of data creation (Carenini et al., 2005).

Impact on study of literary and journalistic texts: Sources of social networks that are primarily textual in nature such as literary texts, historical texts, or news articles are currently under-utilized for social network analysis. In fact, to the best of my knowledge, there is no formal comprehensive categorization of social interactions. An early effort to illustrate the importance of such linkages is by Moretti (2005). In his book, *Graphs, Maps, Trees: Abstract Models for a Literary History*, Moretti presents interesting insights into a novel by looking at its interaction graph. He notes that his models are incomplete because they neither have a notion of weight (number of times two characters interact) nor a notion of direction (mutual or one-directional). There has been recent work that partially addresses these concerns (Elson et al., 2010; Celikyilmaz et al., 2010). They only extract mutual interactions that are signaled by quoted speech. My thesis will go beyond quoted speech and will extract interactions signaled by any linguistic means, in particular verbs of social interaction. Moreover, my research will not only enable extraction of mutual linkages ("who talks to whom") but also of one-directional linkages ("who talks about whom"). This will give rise to new applications such as characterization of literary texts based on the type of social network that underlies the narrative. Moreover, analyses of large amounts of related text such as decades of news articles or historical texts will become possible. By looking at the overall social structure the analyst or scientist will get a summary of the key players and their interactions with each other and the rest of network.

Impact on Linguistics: To the best of my knowledge, there is no cognitive or linguistic theory that explains how people use language to express social interactions. A system that detects lexical items and syntactic constructions that realize interactions and then classifies them into one of the categories, I define in Section 2, has the potential to provide linguists with empirical data to formulate such a theory. For example, the notion of social interactions could be added to the FrameNet resource (Baker and Fillmore, 1998) which is based on frame semantics. FrameNet records possible semantic frames for lexical items. Frames describe lexical meaning by specifying a set of frame elements, which are participants in a typical event or state of affairs expressed by the frame. It provides lexicographic example annotations that illustrate how frames and frame elements can be realized by syntactic constructions. My categorization of social events can be incorporated into FrameNet by adding new frames for social events to the frame hierarchy. The data I collect using the system can provide example sentenctes for these frames. Linguists can use this data to make generalizations about linguistic constructions that realize social interactions frames. For example, a possible generalization could be that transitive verbs in which both subject and object are people, frequently express a social event. In addition, it would be interesting to see what kind social interactions occur in different text genres and if they are realized differently. For example, in a news corpus we hardly found expressions of non-verbal mutual interactions (like eye-contact) while these are frequent in fiction

texts like *Alice in Wonderland*.

2 Work to date

So far, I have defined a comprehensive set of social events and have acquired reliable annotations on a well-known news corpus. I have built a preliminary system that extracts social events from news articles. I will now expand on each of these in the following paragraphs.

Meaning of social events: A text can describe a social network in two ways: explicitly, by stating the type of relationship between two individuals (e.g. *Mary is John's wife*), or implicitly, by describing an event which initiates or perpetuates a social relationship (e.g. *John talked to Mary*). I call the later types of events "social events" (Agarwal et al., 2010). I defined two broad types of social events: **interaction**, in which both parties are aware of each other and of the social event, e.g., a conversation, and **observation**, in which only one party is aware of the other and of the interaction, e.g., thinking of or talking about someone. For example, sentence 1, contains two distinct social events: interaction: *Toujan* was informed by the *committee*, and observation: *Toujan* is talking about the *committee*. I have also defined sub-categories for each of these broad categories based on physical proximity, verbal and non-verbal interactions. For details and examples of these sub-categories please refer to Agarwal et al. (2010)

(1) [Toujan Faisal], 54, {said} [she] was {informed} of the refusal by an [Interior Ministry committee] overseeing election preparations.

As a pilot test to see if creating a social network based on social events can give insight into the social structures of a story, I manually annotated a short version of *Alice in Wonderland*. On the manually extracted network, I ran social network analysis algorithms to answer questions like: who are the most influential characters in the story, which characters have the same social roles and positions. The most influential characters in the story were detected correctly. Another finding was that characters appearing in the same scene like *Dodo*, *Lory*, *Eaglet*, *Mouse* and *Duck* were assigned the same social roles and positions. This pointed out the possibility of using my method to identify separate scenes or sub-plots in a narrative, which is crucial for a better understanding of the text under investigation.

Motivated by this pilot test I decided to annotate social events on the Automatic Content Extraction (ACE) dataset (Doddington et al., 2004), a well known news corpus. My annotations extend previous annotations for entities, relations and events that are present in the 2005 version of the corpus. My annotations revealed that about 80% of the times, entities mentioned together in the same sentence were not linked with any social event. Therefore, a simple heuristic of connecting entities that are present in the same sentence with a link will not reveal a meaningful network. Hence I saw a need for a more sophisticated analysis.

Extraction of social events: To perform such an analysis, I built models for two tasks: social event detection and social event classification (Agarwal and Rambow, 2010). Both were formulated as binary tasks: the first one being about detecting existence of a social event between a pair of entities in a sentence and the second one being about differentiating between the interaction and observation type events (given there is an event between the entities). I used tree kernels on structures derived from phrase structure trees and dependency trees in conjunction with Support Vector Machines (SVMs) to solve the tasks. For the design of structures and type of kernel, I took motivation from a system proposed by Nguyen et al. (2009) which is a state-of-the-art system for relation extraction. I tried all the kernels and their combinations proposed by Nguyen et al. (2009). I used syntactic and semantic insights to devise a new structure derived from dependency trees and showed that this plays a role in achieving the best performance for both social event detection and classification tasks. The reason for choosing such representations is motivated by extensive studies about the regular relation between verb alternations and meaning components (Levin, 1993; Schuler, 2005). This regularity provides a useful generalization that helps to overcome lexical sparseness. However, in order to exploit such regularities, there is a need to have access to a representation which makes the predicate-argument structure clear. Dependency representations do this. Phrase structure representations also represent predicate-argument structure,

but in an indirect way through the structural configurations. These experiments showed that as a result of how language expresses the relevant information, dependency-based structures are best suited for encoding this information. Furthermore, because of the complexity of the task, a combination of phrase-based structures and dependency-based structures perform the best. To my surprise, the system performed extremely well on a seemingly hard task of differentiating between interaction and observation type social events. This result showed that there are significant clues in the lexical and syntactic structures that help in differentiating mutual and one-directional interactions.

3 Future Work

Currently I am working on incorporating semantic resources to improve the performance of my preliminary system. I will work on making convolution kernels scalable and interpretable. These two steps will meet my goal of building a system that will extract social networks from news articles. My next step will be to survey and incorporate domain adaptation techniques that will allow me port my system to other genres like literary and historical texts, blog comments, emails etc. These steps will allow me to extract social networks from a wide range of textual data. At the same time I will be able to empirically analyze the types of linguistic patterns, both lexical and syntactic, that perpetuate social interactions. Now I will expand on the aforementioned future directions.

Adding semantic information: Currently I am exploring linguistically motivated enhancements of dependency and phrase structure trees to formulate new kernels. Specifically, I am exploring ways of incorporating semantic information from VerbNet and FrameNet. This will help me reduce data sparseness and thus improve my current system. I am interested in modeling classes of events which are characterized by the cognitive states of participants– who is aware of whom. The predicate-argument structure of verbs can encode much of this information very efficiently, and classes of verbs express their predicate-argument structure in similar ways. Levin's verb classes, and Palmer's VerbNet (Levin, 1993; Schuler, 2005), are based on syntactic similarity between verbs: two verbs are in the same class if and only if they can realize their arguments in the same syntactic patterns. By the Levin Hypothesis, this is because they share meaning elements, and meaning and syntactic realizations of arguments are related. However, this does not mean that verbs in the same Levin or VerbNet class are synonyms; for example, *to deliberate* and *to play* are both in VerbNet class meet-36.3-1. But from a social event perspective, I am not interested in exact synonymy, and in fact it is quite possible that what I am interested in (awareness of the interaction by the event participants) is the same among verbs of the same VerbNet class. In this case, VerbNet will provide a useful abstraction. Future work will also explore FrameNet, which provides a different type of semantic abstraction and explicit semantic relations that are not directly based on syntactic realizations.

Scaling convolution kernels: Convolution kernels, first proposed by Haussler (1999), are a convenient way of "naturally" combining a variety of features without having to do fine-grained feature engineering. Collins and Duffy (2002) presented a way of successfully using them for NLP tasks such as parsing and tagging. Since then they have been used for various NLP tasks such as relation extraction (Zelenko et al., 2002; Culotta and Jeffrey, 2004; Nguyen et al., 2009), semantic role labeling (Moschitti et al., 2008), question-answer classification (Moschitti et al., 2007) etc. Convolution kernels calculate the similarity between two objects, like trees or strings, by a recursive calculation over the "parts" (substrings, subtrees) of objects. This calculation is usually made computationally efficient by using dynamic programming. But there are two limitations: 1) the computation is still quadratic and hence slow and 2) the features (or parts) that are given high weights at the time of learning remain inaccessible i.e. interpretability of the model becomes difficult.

One direction I will explore to make convolution kernels more scalable is the following: The decision function for the classifier (SVM in dual form) is given in equation 1 (Burges, 1998, Eq 61). In this equation, y_i denotes the class of the i^{th} support vector (s_i), α_i denotes the Lagrange multiplier of s_i, $K(s_i, x)$ denotes the kernel similarity between s_i and a test example x, b denotes the bias. The kernel definition proposed by Collins and Duffy (2002) is given in equation 2, where $h_s(T)$ is the number of

times the s^{th} subtree appears in tree T. The kernel function $K(T_1, T_2)$ therefore calculates the similarity between trees T_1 and T_2 by counting the common subtrees in them. By combining equations 1 and 2 I get equation 3 which can be re-written as equation 4.

$$f(x) = \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b \quad (1)$$

$$K(T_1, T_2) = \sum_{s} h_s(T_1) h_s(T_2) \quad (2)$$

$$f(x) = \sum_{i=1}^{N_s} \alpha_i y_i \sum_{s} h_s(s_i) h_s(x) \quad (3)$$

$$f(x) = \sum_{s} \sum_{i=1}^{N_s} \alpha_i y_i h_s(s_i) h_s(x) \quad (4)$$

The motivation for exchanging these summation signs is that the contribution of larger subtrees to the kernel similarity is strictly less than the contribution of the smaller subtrees. I will investigate the possibility of approximating the decision function of SVM without having to compare all subtrees, in particular large subtrees. I will also investigate if this summation can be calculated in parallel to make the calculation more scalable. Pelossof and Ying (2010) have done recent work on speeding up the Perceptron by stopping the evaluation of features at an early stage if they have high confidence that the example will be classified correctly. Another relevant work to improve the scalability of linear classifiers is due to Clarkson et al. (2010). However, to the best of my knowledge, there is no work that addresses approximation of kernel evaluation for convolution kernels.

Interpretability of convolution kernels: As mentioned in the previous paragraph, another disadvantage of using convolution kernels is that interpretability of a model is difficult. Recently, Pighin and Moschitti (2009) proposed an algorithm to linearize convolution kernels. They show that by efficiently encoding the "relevant" fragments generated by tree kernels, it is possible to get insight into the substructures that were given high weights at the time of learning a model. But their system currently returns thousands of such fragments. I will investigate if there is a way of summarizing these fragments into a meaningful set of syntactic and lexical classes. By doing so I will be able to empirically see what types of linguistic constructs are used by people to express different types of social interactions thus aiding in formulating a theory of how people express social interactions.

Domain adaptation: To be able to extract social networks from literary and historical texts, I will explore domain adaptation techniques. A notable work in this direction is by Daumé III (2007). This work is especially useful for me because Daumé III presents a straightforward kernelized version of his domain adaptation approach which readily fits the machine learning paradigm I am using for my problem. I will explore the literature to see if better domain adaptation techniques have been suggested since then. Domain adaptation will conclude my overall goal of creating a system that can extract social networks from a wide variety of texts. I will then attempt to extract social networks from the increasing amount of text that is becoming machine readable.

Sentiment Analysis:[1] A natural step to try once I have linkages associated with snippets of text is sentiment analysis. I will use my previous work (Agarwal et al., 2009) on contextual phrase-level sentiment analysis to analyze snippets of text and add polarity to social event linkages. Sentiment analysis will make the social network representation even richer by indicating if people are connected with positive, negative or neutral sentiments. This will not only give us information about the protagonists and antagonists in the text but will also affect the analysis of flow of information through the network.

Acknowledgments

This work was funded by NSF grant IIS-0713548. I would like to thank Dr. Owen Rambow and Daniel Bauer for useful discussions and feedback.

References

Apoorv Agarwal and Owen Rambow. 2010. Automatic detection and classification of social events. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Apoorv Agarwal, Fadi Biadsy, and Kathleen Mckeown. 2009. Contextual phrase-level polarity analysis using

[1] I do not mention sentiment analysis anywhere else in my proposal since I will simply use my earlier work.

lexical affect scoring and syntactic n-grams. *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 24–32.

Apoorv Agarwal, Owen C. Rambow, and Rebecca J. Passonneau. 2010. Annotation scheme for social network extraction from text. In *Proceedings of the Fourth Linguistic Annotation Workshop*.

C. Baker and C. Fillmore. 1998. The berkeley framenet project. *Proceedings of the 17th international conference on Computational linguistics*, 1.

Chris Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*.

G. Carenini, R. T. Ng, and X. Zhou. 2005. Scalable discovery of hidden emails from large folders. *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 544–549.

Asli Celikyilmaz, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. 2010. The actor-topic model for extracting social networks in literary narrative. *NIPS Workshop: Machine Learning for Social Computing*.

K. L. Clarkson, E. Hazan, and D. P. Woodruff. 2010. Sublinear optimization for machine learning. *51st Annual IEEE Symposium on Foundations of Computer Science*, pages 449 –457.

M. Collins and N. Duffy. 2002. Convolution kernels for natural language. In *Advances in neural information processing systems*.

Aron Culotta and Sorensen Jeffrey. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 423–429, Barcelona, Spain, July.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. The automatic content extraction (ace) program–tasks, data, and evaluation. *LREC*, pages 837–840.

P. Domingos and M. Richardson. 2003. Mining the network value of customers. *In Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, pages 57–66.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz.

Jianming He, Wesley W. Chu, and Zhenyu (Victor) Liu. 2006. Inferring privacy information from social networks. *Intelligence and Security Informatics*, pages 154–165.

Hal Daume III. 2007. Frustratingly easy domain adaptation. *Annual Meeting-Association For Computational Linguistics*.

D. Kempe, J. Kleinberg, and E. Tardos. 2003. Maximizing the spread of influence through a social network. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation.* The University of Chicago Press.

J. Lindamood, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham. 2009. Inferring private information using social network dataset. *WWW*.

Franco Moretti. 2005. *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso.

A. Moschitti, S. Quarteroni, and R. Basili. 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. *Proceedings of the 45th Conference of the Association for Computational Linguistics (ACL)*.

A. Moschitti, D. Pighin, and R. Basili. 2008. Tree kernels for semantic role labeling. *Computational Linguistics*, 34.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. *Conference on Empirical Methods in Natural Language Processing*.

Raphael Pelossof and Zhiliang Ying. 2010. The attentive perceptron. *CoRR*, abs/1009.5972.

D. Pighin and A. Moschitti. 2009. Reverse engineering of tree kernel feature spaces. *Proceedings of the Conference on EMNLP*, pages 111–120.

Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo. 2007. Automated social hierarchy detection through email network analysis. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 109–117.

Karin Kipper Schuler. 2005. *Verbnet: a broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA. AAI3179808.

D. Zelenko, C. Aone, and A. Richardella. 2002. Kernel methods for relation extraction. In *Proceedings of the EMNLP*.

Elena Zheleva and Lise Getoor. 2009. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. *Proceedings of the 18th international conference on World wide web*, pages 531–540.

Automatic Headline Generation using Character Cross-Correlation

Fahad A. Alotaiby
Department of Electrical Engineering,
College of Engineering, King Saud University
P.O.Box 800, Riyadh 11421, Saudi Arabia
falotaiby@hotmail.com

Abstract

Arabic language is a morphologically complex language. Affixes and clitics are regularly attached to stems which make direct comparison between words not practical. In this paper we propose a new automatic headline generation technique that utilizes character cross-correlation to extract best headlines and to overcome the Arabic language complex morphology. The system that uses character cross-correlation achieves ROUGE-L score of 0.19384 while the exact word matching scores only 0.17252 for the same set of documents.

1 Introduction

A headline is considered as a condensed summary of a document. It can be classified as the acme of text summarization. The necessity for automatic headline generation has been raised due to the need to handle huge amount of documents, which is a tedious and time-consuming process. Instead of reading every document, the headline can be used to decide which of them contains important information.

There are two major disciplines towards automatic headline generation: extractive and abstractive. In the work of (Douzidia and Lapalme, 2004), and extractive method was used to produce a 10-words summary (which can be considered as a headline) of an Arabic document, and then it was automatically translated into English. Therefore, the reported score reflects the accuracy of the generation and translation which makes it difficult to evaluate the process of headline generation of this system. Hedge Trimmer (Dorr *et al.*, 2003) is a system that creates a headline for an English newspaper story using linguistically-motivated heuristics to choose a potential headline. Jin and Hauptmann (2002) proposed a probabilistic model for headline generation in which they divide headline generation process into two steps; namely the step of distilling the information source from the observation of a document and the step of generating a title from the estimated information source, but it was for English documents.

1.1 Headline Length

One of the tasks of the Document Understanding Conference of 2004 (DUC 2004) was generating a very short summary which can be considered as a headline. The evaluation was done on the first 75 bytes of the summary. Knowing that the average word size in Arabic is 5 characters (Alotaiby *et al.* 2009) in addition to space characters, the specified summary size in Arabic words was roughly equivalent to 12 words. In the meantime, the average length of the headlines was about 8 words in the Arabic Gigaword corpus (Graff, 2007) of articles and their headlines. In this work, a 10-words headline is considered as an appropriate length.

1.2 Arabic Language

Classical Arabic writing system was originally consonantal and written from right to left. Every letter in the 28 Arabic alphabets represents a single consonant. To overcome the problem of different pronunciations of consonants in Arabic text, graph-

Proceedings of the ACL-HLT 2011 Student Session, pages 117–121,
Portland, OR, USA 19-24 June 2011.

ical signs known as diacritics were invented in the seventh century. Currently in the Modern Standard Arabic (MSA), diacritics are omitted from written text almost all the time. As a result, this omission increases the number homographs (words with the same writing form). However, Arab readers normally differentiate between homographs by the context of the script.

Moreover, Arabic is a morphologically complex language. An Arabic word may be constructed out of a stem plus affixes and clitics. Furthermore, some parts of the stem may be deleted or modified when appending a clitic to it according to specific orthographical rules. As a final point, different orthographic conventions exist across the Arab world (Buckwalter, 2004). As a result of omitting diacritics, complex morphology and different orthographical rules, two same words may be regarded as different if compared literally.

2 Evaluation Tools

Correctly evaluating the automatically generated headlines is an important phase. Automatic methods for evaluating machine generated headlines are preferred against human evaluations because they are faster, cost effective and can be performed repeatedly. However, they are not trivial because of various factors such as readability of headlines and adequacy of headlines (whether headlines indicate the main content of news story). Hence, it is hard for a computer program to judge. Nevertheless, there are some automatic metrics available for headline evaluation. F1, BLEU (Papineni *et al.* 2002) and ROUGE (Lin, 2004*a*) are the main metrics used.

The evaluation of this experiment was performed using Recall-Oriented Understudy for Gisting Evaluation (ROUGE). ROUGE is a system for measuring the quality of a summary by comparing it to a correct summary created by human. ROUGE provides four different measures, namely ROUGE-*n* (usually *n* = 1,2,3,4), ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. Lin (2004*b*) showed that ROUGE-1, ROUGE-L, ROUGE-SU, and ROUGE-W were very good measures in the category of short summaries.

3 Preparing Data

The dataset used in this work was extracted from Arabic Gigaword (Graff, 2007). The Arabic Gigaword is a collection of text data extracted from newswire archives of Arabic news sources and their titles that have been gathered over several years by the Linguistic Data Consortium (LDC) at the University of Pennsylvania. Text data in the Arabic Gigaword were collected from four newspapers and two press agencies. The Arabic Gigaword corpus contains almost two million documents with nearly 600 million words. For this work, 260 documents were selected from the corpus based on the following steps:

- 3170 documents were selected automatically according to the following:
  i. The length of the document body is between 300 to 1000 words
  ii. The length of the headline (hereafter called original headline) was between 7 to 15 words.
  iii. All words in the original headline must be found in the document body.
- 260 documents were randomly selected from the 3170 documents.

After automatically generating the headlines, 3 native Arabic speaker examiners were hired to evaluate one of the generated headlines as well as the original headline. Also, they were asked to generate 1 headline each for every document. These new 3 headlines will be used as reference headlines in ROUGE to evaluate all automatically generated headlines and the original headline.

4 Headline Extraction Techniques

The main idea of the used method is to extract the most appropriate set of consecutive words (phrase) from a document body that should represent an adequate headline for the document. Then, evaluate those headlines by calculating ROUGE score against a set of 3 reference headlines.

To do so, first, a list of nominated headlines was created from the document body. After this, four different evaluation methods were applied to choose the best headline that reflects the idea of the document among the nominated list. The task of these methods is to catch the most suitable headline that matches the document. The idea here is to

choose the headline that contains the largest number of the most frequent words in the document taking into account ignoring stop words and giving earlier sentences in documents more weight.

4.1 Nominating a List of Headlines

A window of a length of 10-words was passed over the paragraphs word by word to generate chunks of consecutive words that could be used as headlines. Moving the widow one word step may corrupt the fluency of the sentences. A simple approach to reduce this issue is to minimize the size of paragraphs. Therefore, the document body was divided into smaller paragraphs at new-line, comma, colon and period characters. This step increased the number of nominated headlines with proper start and end. The resulting is a nominated list of headlines of a length of 10 words. In the case of a paragraph of a length less than 10, there will be only one nominated headline of the same length of that paragraph.

Table 1 shows an example of nominating headline list where *a* is the selected paragraph, *b* is the first nominated headline and *c* is the second nominated headline. Nominated headlines *b* and *c* are word-by-word translated.

a	ارتبطت نشأة المخطوطات العربية في السودان ببروز معالم الثقافة العربية الإسلامية،
	The emerging of the Arabic manuscripts in Sudan was associated with the rise of the formation of Arabic-Islamic culture,
b	ارتبطت نشأة المخطوطات العربية في السودان ببروز معالم الثقافة العربية
	Associated emerging manuscripts Arabic in Sudan with-rise formation culture Arabic
c	نشأة المخطوطات العربية في السودان ببروز معالم الثقافة العربية الإسلامية
	Emerging manuscripts Arabic in Sudan with-rise formation culture Arabic Islamic

Table 1: An example of headlines nomination.

4.2 Calculating Word Matching Score

The very basic process of making a matching score between every two words in the document body is to give a score of 1 if the two words exactly match or 0 if there is even one mismatch character. This basic step is called the Exact Word Matching (EWM). Unfortunately, Arabic language contains clitics and is morphologically rich. This means the same word could appear with a single clitic attached to it and yet to be considered as a different word in the EWM method. Therefore, the idea of using Character Cross-Correlation (CCC) method emerged. In which a variable score in the range of 0 to 1 is calculated depending on how much characters match with each other. For example, if the word "وكتبها" "*and he wrote it*" is compared with the word "كتب" "*he wrote*" using the EWM method the resulting score will be 0, but when using the CCC method it will be 0.667. The CCC method comes from signals cross-correlation which measures of similarity of two waveforms. In the CCC method the score is calculated according to the following equation:

$$CCC_{w_i,w_j} = \frac{2\max_n c[n]}{M+N} \quad (1)$$

and

$$c[n] = \sum_{m=-(N-1)}^{M-1} w_i[m] * w_j[n+m] \quad (2)$$

where w_i is the first word containing M characters, w_j is the second word containing N characters and the operation * result *1* if the two corresponding characters match each other and *0* otherwise.

4.3 Calculating Best Headline Score

After preparing the two tables of words matching score, now they will be utilized in the selection of the best headline. Except stop-words, every word in the document body (w_d) will be matched with every word in the nominated headline (w_h) using the CCC and the EWM methods and a score will be registered for every nominated sentence. A simple stop-word list consisting of about 180 words was created for this purpose. Calculating matching score for every sentence is also performed in two ways. The first way is the SUM method which is defined in the following equation:

$$SUM_p = \sum_{i=1}^{L} \sum_{j=1}^{K} CCC_{w_d,w_j} \quad (3)$$

where SUM_p is the score using SUM method for the nominated headline p, K is the size of unique words in the document body and L is the size of words in the nominated headline (except stop-words).

In this method the summation of the cross-correlation score of every word in the document body and every word in the headline is added up.

In a similar way, in the other method MAX_p the maximum score between every word in the document body and the nominated headline is added up. Therefore, for every word in the document, its maximum matching score will be added in either cases, CCC or EWM. And it can be defined in the following equation:

$$MAX_p = \sum_{i=1}^{L} \max_j CCC_{w_d,w_j} \quad (4)$$

SUM_p and MAX_p were calculated using EWM and CCC method resulting four different variation of the algorithm namely SUM-EWM, SUM-CCC, MAX-EWM and MAX-CCC.

4.4 Weighing Early Nominated Headlines

In the case of news articles usually the early sentences absorb the subject of the article (Wasson, 1998). To reflect that, a nonlinear multiplicative scaling factor was applied. With this scaling factor, late sentences are penalized. The suggested scaling factor is inspired from sigmoid functions and described in the following equations.

$$SF = -\left(\frac{e^z-1}{e^z+1} - 1\right)/2 \quad (5)$$

where

$$z = 5\left(\frac{2r}{S} - 1\right) \quad (6)$$

and r is the rank of the nominated headline and S is the total number of sentences.

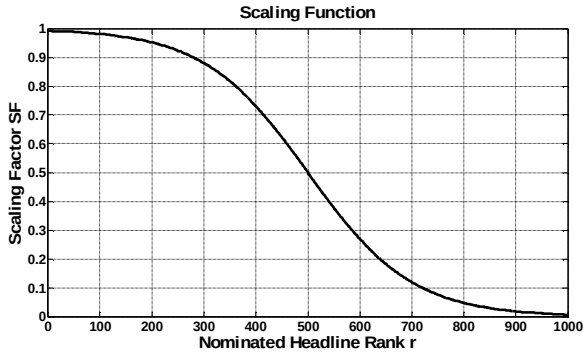

Figure 1: Scaling function of a 1000 nominated headline document.

According the nominating mechanism hundreds of sentences could be nominated as possible headlines. Figure 1 shows the scaling function of a one thousand nominated headlines. After applying the scaling factor, the headline with the maximum score was chosen.

5 Results

Table 2 shows the ROUGE-1 and ROUGE-L scores on the test data. ROUGE-1 measures the co-occurrences of unigrams where ROUGE-L is based on the longest common subsequence (LCS) of an automatically generated headline and the reference headlines.

It is clear that the MAX-CCC scores the highest result in the automatically generated headlines. Unfortunately there are no available results on an Arabic headline generation system to compare with and it is not right to compare these results with other systems applied on other languages or different datasets. So, to give ROUGE score a meaningful aspect, the original headline was evaluated in addition to randomly selected 10 words (Rand-10) and the first 10 words (Lead-10) in the document.

Method	ROUGE-1 (95%-conf.)	ROUGE-L (95%-conf.)
Rand-10	0.08153	0.07081
Lead-10	0.18353	0.17592
SUM-EWM	0.11006	0.10624
SUM-CCC	0.18974	0.17944
MAX-EWM	0.18279	0.17252
MAX-CCC	0.20367	0.19384
Original	0.37683	0.36329

Table 2: ROUGE scores on the test data.

From the registered results it is clear that the MAX-CCC has overcome the problem of the rich existence of clitics and morphology.

6 Conclusions

We have shown the effectiveness of using character cross-correlation in choosing the best headline out of nominated sentences from Arabic document. The advantage of using character cross-correlation is to overcome the complex morphology of the Arabic language. In the comparative experiment, character cross-correlation got ROUGE-L=0.19384 and outperformed the exact word match which got ROUGE-L= 0.17252. Therefore, we conclude that character cross-correlation is effective when com-

paring words in morphologically complex languages such as Arabic.

Acknowledgments

I would like to thank His Excellency the Rector of King Saud University Prof. Abdullah Bin Abdulrahman Alothman for supporting this work by a direct grant. I would also like to thank Dr. Salah Foda and Dr. Ibrahim Alkharashi, my PhD supervisors, for their help in this work.

References

Bonnie Dorr, David Zajic and Richard Schwartz. Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation. In Proceedings of the HLT-NAACL 2003 Text Summarization Workshop and Document Understanding Conference (DUC 2003), Edmonton, Alberta, 2003.

Chin-Yew Lin, ROUGE: a Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out, pages 56-60, Barcelona, Spain, July, 2004*a*.

Chin-Yew Lin, Looking for a few Good Metrics: ROUGE and its Evaluation, In Working Notes of NTCIR-4 (Vol. Supl. 2), 2004*b*.

Document Understanding Conference, http://duc.nist.gov/duc2004/tasks.html, 2004.

Fahad Alotaiby, Ibrahim Alkharashi and Salah Foda. Processing large Arabic text corpora: Preliminary analysis and results. In Proceedings of the Second International Conference on Arabic Language Resources and Tools, pages 78-82, Cairo, Egypt, 2009.

Fouad Douzidia and Guy Lapalme, Lakhas, an Arabic summarization system. In Proceedings of Document Understanding Conference (DUC), Boston, MA, USA, 2004.

David Graff. Arabic Gigaword Third Edition. Linguistic Data Consortium. Philadelphia, USA, 2007.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002.

Mark Wasson. Using Lead Text for news Summaries: Evaluation Results and Implications for Commercial Summarization Applications. In Proceedings of the 17th International Conference on Computational liguistics, Montreal, Canada, 1998.

Rong Jin, and Alex G. Hauptmann, A New Probabilistic Model for Title Generation, The 19th International Conference on Computational Linguistics, Academia Sinica, Taipei, Taiwan, 2002.

Tim Buckwalter. Issues in Arabic Orthography and Morphology Analysis. In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, Geneva, Switzerland, 2004.

Zajic. D., Dorr. B. and Richard Schwartz. Automatic Headline Generation for Newspaper Stories. In Workshop on Automatic Summarization, pages. 78-85, Philadelphia, PA, 2002.

K-means Clustering with Feature Hashing

Hajime Senuma
Department of Computer Science
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
hajime.senuma@gmail.com

Abstract

One of the major problems of K-means is that one must use dense vectors for its centroids, and therefore it is infeasible to store such huge vectors in memory when the feature space is high-dimensional. We address this issue by using feature hashing (Weinberger et al., 2009), a dimension-reduction technique, which can reduce the size of dense vectors while retaining sparsity of sparse vectors. Our analysis gives theoretical motivation and justification for applying feature hashing to K-means, by showing how much will the objective of K-means be (additively) distorted. Furthermore, to empirically verify our method, we experimented on a document clustering task.

1 Introduction

In natural language processing (NLP) and text mining, clustering methods are crucial for various tasks such as document clustering. Among them, K-means (MacQueen, 1967; Lloyd, 1982) is "the most important flat clustering algorithm" (Manning et al., 2008) both for its simplicity and performance.

One of the major problems of K-means is that it has K centroids which are dense vectors where K is the number of clusters. Thus, it is infeasible to store them in memory and slow to compute if the dimension of inputs is huge, as is often the case with NLP and text mining tasks. A well-known heuristic is truncating after the most significant features (Manning et al., 2008), but it is difficult to analyze its effect and to determine which features are significant.

Recently, Weinberger et al. (2009) introduced feature hashing, a simple yet effective and analyzable dimension-reduction technique for large-scale multitask learning. The idea is to combine features which have the same hash value. For example, given a hash function h and a vector $\boldsymbol{x}$, if $h(1012) = h(41234) = 42$, we make a new vector $\boldsymbol{y}$ by setting $y_{42} = x_{1012} + x_{41234}$ (or equally possibly $x_{1012} - x_{41234}$, $-x_{1012} + x_{41234}$, or $-x_{1012} - x_{41234}$).

This trick greatly reduces the size of dense vectors, since the maximum index value becomes equivalent to the maximum hash value of h. Furthermore, unlike random projection (Achlioptas, 2003; Boutsidis et al., 2010), feature hashing retains sparsity of sparse input vectors. An additional useful trait for NLP tasks is that it can save much memory by eliminating an alphabet storage (see the preliminaries for detail). The authors also justified their method by showing that with feature hashing, dot-product is unbiased, and the length of each vector is well-preserved with high probability under some conditions.

Plausibly this technique is useful also for clustering methods such as K-means. In this paper, to motivate applying feature hashing to K-means, we show the residual sum of squares, the objective of K-means, is well-preserved under feature hashing. We also demonstrate an experiment on document clustering and see the feature size can be shrunk into 3.5% of the original in this case.

Proceedings of the ACL-HLT 2011 Student Session, pages 122–126,
Portland, OR, USA 19-24 June 2011.

2 Preliminaries

2.1 Notation

In this paper, $||\cdot||$ denotes the Euclidean norm, and $\langle\cdot,\cdot\rangle$ does the dot product. $\delta_{i,j}$ is the Kronecker's delta, that is, $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.

2.2 *K*-means

Although we do not describe the famous algorithm of K-means (MacQueen, 1967; Lloyd, 1982) here, we remind the reader of its overall objective for later analysis. If we want to group input vectors into K clusters, K-means can surely output clusters $\omega_1, ...\omega_K$ and their corresponding vectors $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K$ such that they locally minimize the residual sum of squares (RSS) which is defined as

$$\sum_{k=1}^{K} \sum_{\boldsymbol{x}\in\omega_k} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2.$$

In the algorithm, $\boldsymbol{\mu}_k$ is made into the mean of the vectors in a cluster ω_k. Hence comes the name K-means.

Note that RSS can be regarded as a metric since the sum of each metric (in this case, squared Euclidean distance) becomes also a metric by constructing a 1-norm product metric.

2.3 Additive distortion

Suppose one wants to embed a metric space (X, d) into another one (X', d') by a mapping ϕ. Its additive distortion is the infimum of ϵ which, for any observed $x, y \in X$, satisfies the following condition:

$$d(x, y) - \epsilon \leq d'(\phi(x), \phi(y)) \leq d(x, y) + \epsilon.$$

2.4 Hashing tricks

According to an account by John Langford [1], a co-author of papers on feature hashing (Shi et al., 2009; Weinberger et al., 2009), hashing tricks for dimension-reduction were implemented in various machine learning libraries including Vowpal Wabbit, which he realesed in 2007.

Ganchev and Dredze (2008) named their hashing trick *random feature mixing* and empirically supported it by experimenting on NLP tasks. It is similar to feature hashing except lacking of a binary hash function. The paper also showed that hashing tricks are useful to eliminate alphabet storage.

Shi et al. (2009) suggested *hash kernel*, that is, dot product on a hashed space. They conducted thorough research both theoretically and experimentally, extending this technique to classification of graphs and multi-class classification. Although they tested K-means in an experiment, it was used for classification but not for clustering.

Weinberger et al. (2009) [2] introduced a technique *feature hashing* (a function itself is called the *hashed feature map*), which incorporates a binary hash function into hashing tricks in order to guarantee the hash kernel is unbiased. They also showed applications to various real-world applications such as multitask learning and collaborative filtering. Though their proof for exponential tail bounds in the original paper was refuted later, they reproved it under some extra conditions in the latest version. Below is the definition.

Definition 2.1. Let S be a set of hashable features, h be a hash function $h : S \to \{1, ..., m\}$, and ξ be $\xi : S \to \{\pm 1\}$. The *hashed feature map* $\phi^{(h,\xi)} : \mathbb{R}^{|S|} \to \mathbb{R}^m$ is a function such that the i-th element of $\phi^{(h,\xi)}(\boldsymbol{x})$ is given by

$$\phi_i^{(h,\xi)}(\boldsymbol{x}) = \sum_{j:h(j)=i} \xi(j)x_j.$$

If h and ξ are clear from the context, we simply write $\phi^{(h,\xi)}$ as ϕ.

As well, a kernel function is defined on a hashed feature map.

Definition 2.2. The *hash kernel* $\langle\cdot,\cdot\rangle_\phi$ is defined as

$$\langle\boldsymbol{x}, \boldsymbol{x}'\rangle_\phi = \langle\phi(\boldsymbol{x}), \phi(\boldsymbol{x}')\rangle.$$

They also proved the following theorem, which we use in our analysis.

Theorem 2.3. *The hash kernel is unbiased, that is,*

$$\mathbb{E}_\phi[\langle\boldsymbol{x}, \boldsymbol{x}'\rangle_\phi] = \langle\boldsymbol{x}, \boldsymbol{x}'\rangle.$$

The variance is

$$Var_\phi[\langle\boldsymbol{x}, \boldsymbol{x}'\rangle_\phi] = \frac{1}{m}\left(\sum_{i\neq j} x_i^2 x_j'^2 + x_i x_i' x_j x_j'\right).$$

[1] http://hunch.net/~jl/projects/hash_reps/index.html

[2] The latest version of this paper is at arXiv http://arxiv.org/abs/0902.2206, with correction to Theorem 3 in the original paper included in the Proceeding of ICML '09.

2.4.1 Eliminating alphabet storage

In this kind of hashing tricks, an index of inputs do not have to be an integer but can be any hashable value, including a string. Ganchev and Dredze (2008) argued this property is useful particularly for implementing NLP applications, since we do not anymore need an *alphabet*, a dictionary which maps features to parameters.

Let us explain in detail. In NLP, features can be often expediently expressed with strings. For instance, a feature 'the current word ends with -ing' can be expressed as a string `cur:end:ing` (here we suppose `:` is a control character). Since indices of dense vectors (which may be implemented with arrays) must be integers, traditionally we need a dictionary to map these strings to integers, which may waste much memory. Feature hashing removes this memory waste by converting strings to integers with on-the-fly computation.

3 Method

For dimension-reduction to K-means, we propose a new method *hashed K-means*. Suppose you have N input vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$. Given a hashed feature map ϕ, hashed K-means runs K-means on $\phi(\boldsymbol{x}_1), ..., \phi(\boldsymbol{x}_N)$ instead of the original ones.

4 Analysis

In this section, we show clusters obtained by the hashed K-means are also good clusters in the original space with high probability. While Weinberger et al. (2009) proved a theorem on (multiplicative) distortion for Euclidean distance under some tight conditions, we illustrate (additive) distortion for RSS. Since K-means is a process which monotonically decreases RSS in each step, if RSS is not distorted so much by feature hashing, we can expect results to be reliable to some extent.

Let us define the difference of the residual sum of squares (DRSS).

Definition 4.1. Let $\omega_1, ...\omega_K$ be clusters, $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K$ be their corresponding centroids in the original space, ϕ be a hashed feature map, and $\boldsymbol{\mu}_1^\phi, ..., \boldsymbol{\mu}_K^\phi$ be their corresponding centroids in the hashed space. Then, DRSS is defined as follows:

$$\begin{aligned} DRSS &= |\sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \omega_k} ||\phi(\boldsymbol{x}) - \boldsymbol{\mu}_k^\phi||^2 \\ &- \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in \omega_k} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2|. \end{aligned}$$

Before analysis, we define a notation for the (Euclidean) length under a hashed space:

Definition 4.2. The *hash length* $||\cdot||_\phi$ is defined as

$$\begin{aligned} ||\boldsymbol{x}||_\phi &= ||\phi(\boldsymbol{x})|| \\ &= \sqrt{\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}) \rangle} = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_\phi}. \end{aligned}$$

Note that it is clear from Theorem 2.3 that $\mathbb{E}_\phi[||\boldsymbol{x}||_\phi^2] = ||\boldsymbol{x}||^2$, and equivalently $\mathbb{E}_\phi[||\boldsymbol{x}||_\phi^2 - ||\boldsymbol{x}||^2] = 0$.

In order to show distortion, we want to use Chebyshev's inequality. To this end, it is vital to know the expectation and variance of the sum of squared hash lengths. Because the variance of the sum of random variables derives from each covariance between pairs of variables, first we show the covariance between the squared hash length of two vectors.

Lemma 4.3. *The covariance between the squared hash length of two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ is*

$$Cov_\phi(||\boldsymbol{x}||_\phi^2, ||\boldsymbol{y}||_\phi^2) = \frac{\psi(\boldsymbol{x}, \boldsymbol{y})}{m},$$

where

$$\psi(\boldsymbol{x}, \boldsymbol{y}) = 2 \sum_{i \neq j} x_i x_j y_i y_j.$$

This lemma can be proven by the same technique described in the Appendix A of Weinberger et al. (2009).

Now we see the following lemma.

Lemma 4.4. *Suppose we have N vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$. Let us define $X = \sum_i ||\boldsymbol{x}_i||_\phi^2 - \sum_i ||\boldsymbol{x}_i||^2 = \sum_i \left(||\boldsymbol{x}_i||_\phi^2 - ||\boldsymbol{x}_i||^2 \right)$. Then, for any $\epsilon > 0$,*

$$P\left(|X| \geq \frac{\epsilon}{\sqrt{m}} \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} \psi(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right) \leq \frac{1}{\epsilon^2}.$$

Proof. This is an application of Chebyshev's inequality. Namely, for any $\epsilon > 0$,

$$P\left(|X - \mathbb{E}_\phi[X]| \geq \epsilon\sqrt{Var_\phi[X]}\right) \leq \frac{1}{\epsilon^2}.$$

Since the expectation of a sum is the sum of expectations we readily know the zero expectation: $\mathbb{E}_\phi[X] = 0$.

Since adding constants to the inputs of covariance does not change its result, from Lemma 4.3, for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$,

$$Cov_\phi(||\boldsymbol{x}||_\phi^2 - ||\boldsymbol{x}||^2, ||\boldsymbol{y}||_\phi^2 - ||\boldsymbol{y}||^2) = \frac{\psi(\boldsymbol{x}, \boldsymbol{y})}{m}.$$

Because the variance of the sum of random variables is the sum of the covariances between every pair of them,

$$Var_\phi[X] = \frac{1}{m}\sum_{i=1}^{N}\sum_{j=1}^{N}\psi(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

□

Finally, we see the following theorem for additive distortion.

Theorem 4.5. *Let Ψ be the sum of $\psi(\boldsymbol{x}, \boldsymbol{y})$ for any observed pair of $\boldsymbol{x}, \boldsymbol{y}$, each of which expresses the difference between an example and its corresponding centroid. Then, for any ϵ,*

$$P\left(|DRSS| \geq \epsilon\right) \leq \frac{\Psi}{\epsilon^2 m}.$$

Thus, if $m \geq \gamma^{-1}\Psi\epsilon^{-2}$ where $0 < \gamma <= 1$, with probability at least $1-\gamma$, RSS is additively distorted by ϵ.

Proof. Note that a hashed feature map $\phi^{(h,\xi)}$ is linear, since $\phi(\boldsymbol{x}) = M\boldsymbol{x}$ with a matrix M such that $M_{i,j} = \xi(i)\delta_{h(i),j}$. By this liearlity, $\boldsymbol{\mu}_k^\phi = |\omega_k|^{-1}\sum_{\boldsymbol{x}\in\omega_k}\phi(\boldsymbol{x}) = \phi(|\omega_k|^{-1}\sum_{\boldsymbol{x}\in\omega_k}\boldsymbol{x}) = \phi(\boldsymbol{\mu}_k)$. Reapplying linearlity to this result, we have $||\phi(\boldsymbol{x}) - \boldsymbol{\mu}_k^\phi||^2 = ||\boldsymbol{x} - \boldsymbol{\mu}_k||_\phi^2$. Lemma 4.4 completes the proof. □

The existence of Ψ in the theorem suggests that to use feature hashing, we should remove useless features which have high values from data in advance. For example, if frequencies of words are used as features, function words should be ignored not only because they give no information for clustering but also because their high frequencies magnify distortion.

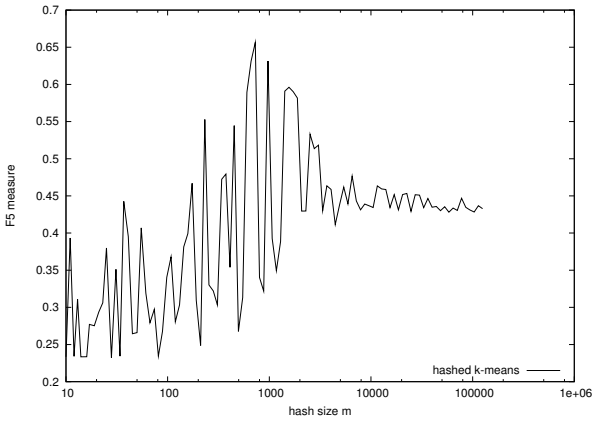

Figure 1: The change of F5-measure along with the hash size

5 Experiments

To empirically verify our method, from 20 Newsgroups, a dataset for document classification or clustering [3], we chose 6 classes and randomly drew 100 documents for each class.

We used unigrams and bigrams as features and ran our method for various hash sizes m (Figure 1). The number of unigrams is 33,017 and bigrams 109,395, so the feature size in the original space is 142,412.

To measure performance, we used the F_5 measure (Manning et al., 2008). The scheme counts correctness pairwisely. For example, if a document pair in an output cluster is actually in the same class, it is counted as true positive. In contrast, if it is actually in the different class, it is counted as false positive. Following this manner, a contingency table can be made as follows:

	Same cluster	Diff. clusters
Same class	TP	FN
Diff. classes	FP	TN

Now, F_β measure can be defined as

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

where the precision $P = TP/(TP+FP)$ and the recall $R = TP/(TP+FN)$.

[3] http://people.csail.mit.edu/jrennie/20Newsgroups/

In short, F_5 measure strongly favors precision to recall. Manning et al. (2008) stated that in some cases separating similar documents is more unfavorable than putting dissimilar documents together, and in such cases the F_β measure (where $\beta > 1$) is a good evaluation criterion.

At the first look, it seems odd that performance can be higher than the original where m is low. A possible hypothesis is that since K-means only **locally** minimizes RSS but in general there are many local minima which are far from the global optimal point, therefore distortion can be sometimes useful to escape from a bad local minimum and reach a better one. As a rule, however, large distortion kills clustering performance as shown in the figure.

Although clustering is heavily case-dependent, in this experiment, the resulting clusters are still reliable where the hash size is 3.5% of the original feature space size (around 5,000).

6 Future Work

Arthur and Vassilvitskii (2007) proposed K-means++, an improved version of K-means which guarantees its RSS is upper-bounded. Combining their method and the feature hashing as shown in our paper will produce a new efficient method (possibly it can be named *hashed K-means++*). We will analyze and experiment with this method in the future.

7 Conclusion

In this paper, we argued that applying feature hashing to K-means is beneficial for memory-efficiency. Our analysis theoretically motivated this combination. We supported our argument and analysis by an experiment on document clustering, showing we could safely shrink memory-usage into 3.5% of the original in our case. In the future, we will analyze the technique on other learning methods such as K-means++ and experiment on various real-data NLP tasks.

Acknowledgements

We are indebted to our supervisors, Jun'ichi Tsujii and Takuya Matsuzaki. We are also grateful to the anonymous reviewers for their helpful and thoughtful comments.

References

Dimitris Achlioptas. 2003. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, June.

David Arthur and Sergei Vassilvitskii. 2007. k-means++ : The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035.

Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. 2010. Random Projections for k-means Clustering. In *Advances in Neural Information Processing Systems 23*, number iii, pages 298–306.

Kuzman Ganchev and Mark Dredze. 2008. Small Statistical Models by Random Feature Mixing. In *Proceedings of the ACL08 HLT Workshop on Mobile Language Processing*, pages 19–20.

Stuart P. Lloyd. 1982. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

J MacQueen. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and S.V.N. Vishwanathan. 2009. Hash Kernels for Structured Data. *Journal of Machine Learning Research*, 10:2615–2637.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature Hashing for Large Scale Multitask Learning. In *Proceedings of the 26th International Conference on Machine Learning*.

Author Index