HLT-NAACL 2006

Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics

Short Papers

Robert C. Moore, General Chair
Jeff Bilmes, Jennifer Chu-Carroll and Mark Sanderson
Program Committee Chairs

June 4-9, 2006
New York, New York, USA

Published by the Association for Computational Linguistics
http://www.aclweb.org

Production and Manufacturing by
Omnipress Inc.
2600 Anderson Street
Madison, WI 53704

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

Table of Contents

Factored Neural Language Models

Andrei Alexandrescu
Department of Comp. Sci. Eng.
University of Washington
andrei@cs.washington.edu

Katrin Kirchhoff
Department of Electrical Engineering
University of Washington
katrin@ee.washington.edu

Abstract

We present a new type of neural probabilistic language model that learns a mapping from both words and explicit word features into a continuous space that is then used for word prediction. Additionally, we investigate several ways of deriving continuous word representations for unknown words from those of known words. The resulting model significantly reduces perplexity on sparse-data tasks when compared to standard backoff models, standard neural language models, and factored language models.

1 Introduction

Neural language models (NLMs) (Bengio et al., 2000) map words into a continuous representation space and then predict the probability of a word given the continuous representations of the preceding words in the history. They have previously been shown to outperform standard back-off models in terms of perplexity and word error rate on medium and large speech recognition tasks (Xu et al., 2003; Emami and Jelinek, 2004; Schwenk and Gauvain, 2004; Schwenk, 2005). Their main drawbacks are computational complexity and the fact that only distributional information (word context) is used to generalize over words, whereas other word properties (e.g. spelling, morphology etc.) are ignored for this purpose. Thus, there is also no principled way of handling out-of-vocabulary (OOV) words. Though this may be sufficient for applications that use a closed vocabulary, the current trend of porting systems to a wider range of languages (esp. highly-inflected languages such as Arabic) calls for dynamic dictionary expansion and the capability of assigning probabilities to newly added words without having seen them in the training data. Here, we introduce a novel type of NLM that improves generalization by using vectors of word features (stems, affixes, etc.) as input, and we investigate deriving continuous representations for unknown words from those of known words.

2 Neural Language Models

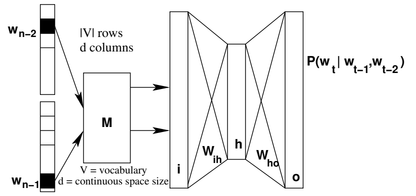


Figure 1: *NLM architecture. Each word in the context maps to a row in the matrix M. The output is next word's probability distribution.*

A standard NLM (Fig. 1) takes as input the previous $n-1$ words, which select rows from a continuous word representation matrix M. The next layer's input $\mathbf{i}$ is the concatenation of the rows in M corresponding to the input words. From here, the network is a standard multi-layer perceptron with hidden layer $\mathbf{h} = \tanh(\mathbf{i} * W_{ih} + \mathbf{b}_h)$ and output layer $\mathbf{o} = \mathbf{h} * W_{ho} + \mathbf{b}_o$. where $\mathbf{b}_{h,o}$ are the biases on the respective layers. The vector $\mathbf{o}$ is normalized by the softmax function $f_{softmax}(o_i) = \frac{e^{o_i}}{\sum_{k=1}^{|V|} e^{o_k}}$. Backpropagation (BKP) is used to learn model parame-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 1–4,
New York, June 2006.

ters, including the M matrix, which is shared across input words. The training criterion maximizes the regularized log-likelihood of the training data.

3 Generalization in Language Models

An important task in language modeling is to provide reasonable probability estimates for n-grams that were not observed in the training data. This generalization capability is becoming increasingly relevant in current large-scale speech and NLP systems that need to handle unlimited vocabularies and domain mismatches. The smooth predictor function learned by NLMs can provide good generalization if the test set contains n-grams whose individual words have been seen in similar context in the training data. However, NLMs only have a simplistic mechanism for dealing with words that were not observed at all: OOVs in the test data are mapped to a dedicated class and are assigned the singleton probability when predicted (i.e. at the output layer) and the features of a randomly selected singleton word when occurring in the input. In standard backoff n-gram models, OOVs are handled by reserving a small fixed amount of the discount probability mass for the generic OOV word and treating it as a standard vocabulary item. A more powerful backoff strategy is used in factored language models (FLMs) (Bilmes and Kirchhoff, 2003), which view a word as a vector of word features or "factors": $w = \langle f_1, f_2, \ldots, f_k \rangle$ and predict a word jointly from previous words and their factors: A generalized backoff procedure uses the factors to provide probability estimates for unseen n-grams, combining estimates derived from different backoff paths. This can also be interpreted as a generalization of standard class-based models (Brown et al., 1992). FLMs have been shown to yield improvements in perplexity and word error rate in speech recognition, particularly on sparse-data tasks (Vergyri et al., 2004) and have also outperformed backoff models using a linear decomposition of OOVs into sequences of morphemes. In this study we use factors in the input encoding for NLMs.

4 Factored Neural Language Models

NLMs define word similarity solely in terms of their context: words are assumed to be close in the continuous space if they co-occur with the same (subset of) words. But similarity can also be derived from word shape features (affixes, capitalization, hyphenation etc.) or other annotations (e.g. POS classes). These allow a model to generalize across classes of words bearing the same feature. We thus define a *factored neural language model* (FNLM) (Fig. 2) which takes as input the previous $n-1$ vectors of factors. Different factors map to disjoint row sets of the matrix. The **h** and **o** layers are identical to the standard NLM's. Instead of predicting the probabilities for all words at the output layer directly, we first group words into classes (obtained by Brown clustering) and then compute the conditional probability of each word given its class: $P(w_t) = P(c_t) \times P(w_t|c_t)$. This is a speed-up technique similar to the hierarchical structuring of output units used by (Morin and Bengio, 2005), except that we use a "flat" hierarchy. Like the standard NLM, the network is trained to maximize the log-likelihood of the data. We use BKP with cross-validation on the development set and L2 regularization (the sum of squared weight values penalized by a parameter λ) in the objective function.

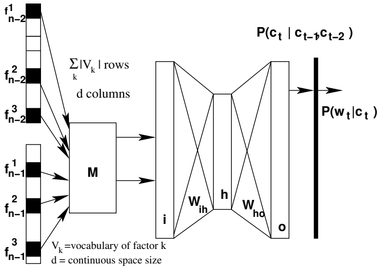


Figure 2: *FNLM architecture. Input vectors consisting of word and feature indices are mapped to rows in M. The final multiplicative layer outputs the word probability distribution.*

5 Handling Unknown Factors in FNLMs

In an FNLM setting, a subset of a word's factors may be known or can be reliably inferred from its shape although the word itself never occurred in the training data. The FNLM can use the continuous representation for these known factors directly in the input. If unknown factors are still present, new continuous representations are derived for them from those of known factors of the same type. This is done by averaging over the continuous vectors of a selected subset of the words in the training data, which places the new item in the center of the region occupied by the subset. For example, proper nouns constitute a large fraction of OOVs, and using the mean of the rows in M associated with words with a proper noun tag yields the "average proper noun" representation for the unknown word. We have experimented with the following strategies for subset selection: NULL (the null subset, i.e. the feature vector components for unknown factors are 0), ALL (average of all known factors of the same type); TAIL (averaging over the least frequently encountered factors of that type up to a threshold of 10%); and LEAST, i.e. the representation of the single least frequent factors of the same type. The prediction of OOVs themselves is unaffected since we use a factored encoding only for the input, not for the output (though this is a possibility for future work).

6 Data and Baseline Setup

We evaluate our approach by measuring perplexity on two different language modeling tasks. The first is the LDC CallHome Egyptian Colloquial Arabic (ECA) Corpus, consisting of transcriptions of phone conversations. ECA is a morphologically rich language that is almost exclusively used in informal spoken communication. Data must be obtained by transcribing conversations and is therefore very sparse. The present corpus has 170K words for training ($|V| = 16026$), 32K for development (dev), 17K for evaluation (eval97). The data was preprocessed by collapsing hesitations, fragments, and foreign words into one class each. The corpus was further annotated with morphological information (stems, morphological tags) obtained from the LDC ECA lexicon. The OOV rates are 8.5% (development set) and 7.7% (eval97 set), respectively.

Model	ECA ($\cdot 10^2$)		Turkish ($\cdot 10^2$)	
	dev	**eval**	**dev**	**eval**
baseline 3gram	4.108	4.128	6.385	6.438
hand-optimized FLM	4.440	4.327	4.269	4.479
GA-optimized FLM	4.325	4.179	6.414	6.637
NLM 3-gram	4.857	4.581	4.712	4.801
FNLM-NULL	5.672	5.381	9.480	9.529
FNLM-ALL	5.691	5.396	9.518	9.555
FNLM-TAIL 10%	5.721	5.420	9.495	9.540
FNLM-LEAST	5.819	5.479	10.492	10.373

Table 1: *Average probability (scaled by 10^2) of known words with unknown words in order-2 context*

The second corpus consists of Turkish newspaper text that has been morphologically annotated and disambiguated (Hakkani-Tür et al., 2002), thus providing information about the word root, POS tag, number and case. The vocabulary size is 67510 (relatively large because Turkish is highly agglutinative). 400K words are used for training, 100K for development (11.8% OOVs), and 87K for testing (11.6% OOVs). The corpus was preprocessed by removing segmentation marks (titles and paragraph boundaries).

7 Experiments and Results

We first investigated how the different OOV handling methods affect the average probability assigned to words with OOVs in their context. Table 1 shows that average probabilities increase compared to the strategy described in Section 3 as well as other baseline models (standard backoff trigrams and FLM, further described below), with the strongest increase observed for the scheme using the least frequent factor as an OOV factor model. This strategy is used for the models in the following perplexity experiments.

We compare the perplexity of word-based and factor-based NLMs with standard backoff trigrams, class-based trigrams, FLMs, and interpolated models. Evaluation was done with (the "w/unk" column in Table 2) and without (the "no unk" column) scoring of OOVs, in order to assess the usefulness of our approach to applications using closed vs. open vocabularies. The baseline Model 1 is a standard backoff 3-gram using modified Kneser-Ney smoothing (model orders beyond 3 did not improve perplexity). Model 2 is a class-based trigram model with Brown clustering (256 classes), which, when interpolated with the baseline 3-gram, reduces the perplexity (see row 3). Model 3 is a 3-gram word-based NLM (with output unit clustering). For NLMs, higher model orders gave improvements, demonstrating their better scalability: for ECA, a 6-gram (w/o unk) and a 5-gram (w/unk) were used; for Turkish, a 7-gram (w/o unk) and a 5-gram (w/unk) were used. Though worse in isolation, the word-based NLMs reduce perplexity considerably when interpolated with Model 1. The FLM baseline is a hand-optimized 3-gram FLM (Model 5); we also tested an FLM optimized with a genetic algorithm as de-

#	Model	ECA dev		ECA eval		Turkish dev		Turkish eval	
		no unk	w/unk	no unk	w/unk	no unk	w/unk	no unk	w/unk
1	Baseline 3-gram	191	176	183	172	827	569	855	586
2	Class-based LM	221	278	219	269	1642	1894	1684	1930
3	1) & 2)	183	169	178	167	790	540	814	555
4	Word-based NLM	208	341	204	195	1510	1043	1569	1067
5	1) & 4)	178	165	173	162	758	542	782	557
6	Word-based NLM	202	194	204	192	1991	1369	2064	1386
7	1) & 6)	175	162	173	160	754	563	772	580
8	hand-optimized FLM	187	171	178	166	827	595	854	614
9	1) & 8)	182	167	174	163	805	563	832	581
10	genetic FLM	190	188	181	188	761	1181	776	1179
11	1) & 10)	183	166	175	164	706	488	720	498
12	factored NLM	189	173	190	175	1216	808	1249	832
13	1) & 12)	169	155	168	155	724	487	744	500
14	1) & 10) & 12)	**165**	**155**	**165**	**154**	**652**	**452**	**664**	**461**

Table 2: *Perplexities for baseline backoff LMs, FLMs, NLMs, and LM interpolation*

scribed in (Duh and Kirchhoff, 2004) (Model 6). Rows 7-10 of Table 2 display the results. Finally, we trained FNLMs with various combinations of factors and model orders. The combination was optimized by hand on the dev set and is therefore most comparable to the hand-optimized FLM in row 8. The best factored NLM (Model 7) has order 6 for both ECA and Turkish. It is interesting to note that the best Turkish FNLM uses only word factors such as morphological tag, stem, case, etc. but not the actual words themselves in the input. The FNLM outperforms all other models in isolation except the FLM; its interpolation with the baseline (Model 1) yields the best result compared to all previous interpolated models, for both tasks and both the unk and no/unk condition. Interpolation of Model 1, FLM and FNLM yields a further improvement. The parameter values of the (F)NLMs range between 32 and 64 for d, 45-64 for the number of hidden units, and 362-1024 for C (number of word classes at the output layer).

8 Conclusion

We have introduced FNLMs, which combine neural probability estimation with factored word representations and different ways of inferring continuous word features for unknown factors. On sparse-data Arabic and Turkish language modeling task FNLMs were shown to outperform all comparable models (standard backoff 3-gram, word-based NLMs) except FLMs in isolation, and all models when interpolated with the baseline. These conclusions apply to both open and closed vocabularies.

Acknowledgments

This work was funded by NSF under grant no. IIS-0326276 and DARPA under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of these agencies.

References

Y. Bengio, R. Ducharme, and P. Vincent. 2000. A neural probabilistic language model. In *NIPS*.

J.A. Bilmes and K. Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *HLT-NAACL*.

P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4).

K. Duh and K. Kirchhoff. 2004. Automatic learning of language model structure. In *COLING 2004*.

A. Emami and F. Jelinek. 2004. Exact training of a neural syntactic language model. In *ICASSP 2004*.

D. Hakkani-Tür, K. Oflazer, and G. Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Journal of Computers and Humanities*, 36(4).

F. Morin and Y. Bengio. 2005. Hierarchical probabilistic neural network language model. In *AISTATS*.

H. Schwenk and J.L. Gauvain. 2004. Neural network language models for conversational speech recognition. In *ICSLP 2004*.

H. Schwenk. 2005. Training neural network language models on very large corpora. In *HLT/EMNLP*.

D. Vergyri, K. Kirchhoff, K. Duh, and A. Stolcke. 2004. Morphology-based language modeling for arabic speech recognition. In *ICSLP*.

P. Xu, A. Emami, and F. Jelinek. 2003. Training connectionist models for the structured language model. In *EMNLP 2003*.

The MILE Corpus for Less Commonly Taught Languages

Alison Alvarez, Lori Levin, Robert Frederking, Simon Fung, Donna Gates
Language Technologies Institute
5000 Forbes Avenue
Pittsburgh, PA 15213
[nosila, lsl, ref+, sfung, dmg]@cs.cmu.edu

Jeff Good
Max Planck Institute for Evolutionary Anthropology
Deutscher Platz 6
04103 Leipzig
good@eva.mpg.de

Abstract

This paper describes a small, structured English corpus that is designed for translation into Less Commonly Taught Languages (LCTLs), and a set of re-usable tools for creation of similar corpora.[1] The corpus systematically explores meanings that are known to affect morphology or syntax in the world's languages. Each sentence is associated with a feature structure showing the elements of meaning that are represented in the sentence. The corpus is highly structured so that it can support machine learning with only a small amount of data. As part of the REFLEX program, the corpus will be translated into multiple LCTLs, resulting in parallel corpora can be used for training of MT and other language technologies. Only the untranslated English corpus is described in this paper.

1 Introduction

Of the 6,000 living languages in the world only a handful have the necessary monolingual or bilingual resources to build a working statistical or example-based machine translation system. Currently, there are efforts to build *language packs* for Less Commonly Taught Languages (LCTLs). Each language pack includes parallel corpora consisting of naturally occurring text translated from English into the LCTL or vice versa.

This paper describes a small corpus that supplements naturally occurring text with highly systematic enumeration of meanings that are known to affect morphology and syntax in the world's languages. The supplemental corpus will enable the exploration of constructions that are sparse or obscured in natural data. The corpus consists of 12,875 English sentences, totaling 76,202 word tokens.

This paper describes the construction of the corpus, including tools and resources that can be used for the construction of similar corpora.

2 Structure of the corpus

- 247: John said "The woman is a teacher."
- 248: John said the woman is not a teacher.
- 249: John said "The woman is not a teacher."
- 250: John asked if the woman is a teacher.
- 251: John asked "Is the woman a teacher?"
- 252: John asked if the woman is not a teacher.
- …
- 1488: Men are not baking cookies.
- 1489: The women are baking cookies.
- …
- 1537: The ladies' waiter brought appetizers.
- 1538: The ladies' waiter will bring appetizers.

Figure 1: A sampling of sentences from the complete elicitation corpus

[1] AVENUE/MILE is supported by the US National Science Foundation NSF grant number IIS-0121-631 and the US Government's REFLEX Program.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 5–8,
New York, June 2006. ©2006 Association for Computational Linguistics

```
srcsent: Mary was not a doctor.
context: Translate this as though it were spoken to a peer co-worker;

((actor ((np-function fn-actor)(np-animacy anim-human)(np-biological-gender bio-gender-female)
        (np-general-type   proper-noun-type)(np-identifiability identifiable)
        (np-specificity specific)…))
(pred ((np-function fn-predicate-nominal)(np-animacy anim-human)(np-biological-gender bio-
        gender-female) (np-general-type common-noun-type)(np-specificity specificity-neutral)…))
(c-v-lexical-aspect state)(c-copula-type copula-role)(c-secondary-type secondary-copula)(c-
solidarity solidarity-neutral) (c-power-relationship power-peer) (c-v-grammatical-aspect gram-
aspect-neutral)(c-v-absolute-tense past) (c-v-phase-aspect phase-aspect-neutral) (c-general-
type declarative-clause)(c-polarity polarity-negative)(c-my-causer-intentionality intentionality-
n/a)(c-comparison-type comparison-n/a)(c-relative-tense relative-n/a)(c-our-boundary boundary-
n/a)…)
```

Figure 2: An abridged feature structure, sentence and context field

The MILE (Minor Language Elicitation) corpus is a highly structured set of English sentences. Each sentence represents a meaning or combination of meanings that we want to elicit from a speaker of an LCTL. For example, the corpus excerpts in Figure 1 explore quoted and non quoted sentential complements, embedded questions, negation, definiteness, biological gender, and possessive noun phrases.

Underlying each sentence is a feature structure that serves to codify its meaning. Additionally, sentences are accompanied by a context field that provides information that may be present in the feature structure, but not inherent in the English sentence. For example, in Figure 2, the feature structure specifies solidarity with the hearer and power relationship of the speaker and hearer, as evidenced by the features-value pairs *(c-solidarity solidarity-neutral)* and *(c-power-relationship power-peer)*. Because this is not an inherent part of English grammar, this aspect of meaning is conveyed in the context field.

3 Building the Corpus

Figure 3 shows the steps in creating the corpus. Corpus creation is driven by a Feature Specification. The Feature Specification defines features such as tense, person, and number, and values for each feature such past, present, future, remote past, recent past, for tense. Additionally, the feature specification defines illegal combinations of features, such as the use of a singular number with an inclusive or exclusive pronoun (*We* = *you* and *me* vs *we* = *me* and *other people*). The inventory of features and values is informed by typological studies of which elements of meaning are known to affect syntax and morphology in some of the world's languages. The feature specification currently contains 42 features and 340 values and covers. In order to select the most relevant features we drew guidance from Comrie and Smith (1977) and Bouquiaux and Thomas (1992). We also used the *World Atlas of Language Structures* (Haspelmath et al. 2005) as a catalog of existing language features and their prevalence.

In the process of corpus creation, feature structures are created before their corresponding English sentences. There are three reasons for this. First, as mentioned above, the feature structure may contain elements of meaning that are not explicitly represented in the English sentence. Second, multiple elicitation languages can be generated from the same set of feature structures. For example, when we elicit South American languages we use Spanish instead of English sentences. Third, what we want to know about each LCTL is not how it translates the structural elements of English such as determiners and auxiliary verbs, but how it renders certain meanings such as

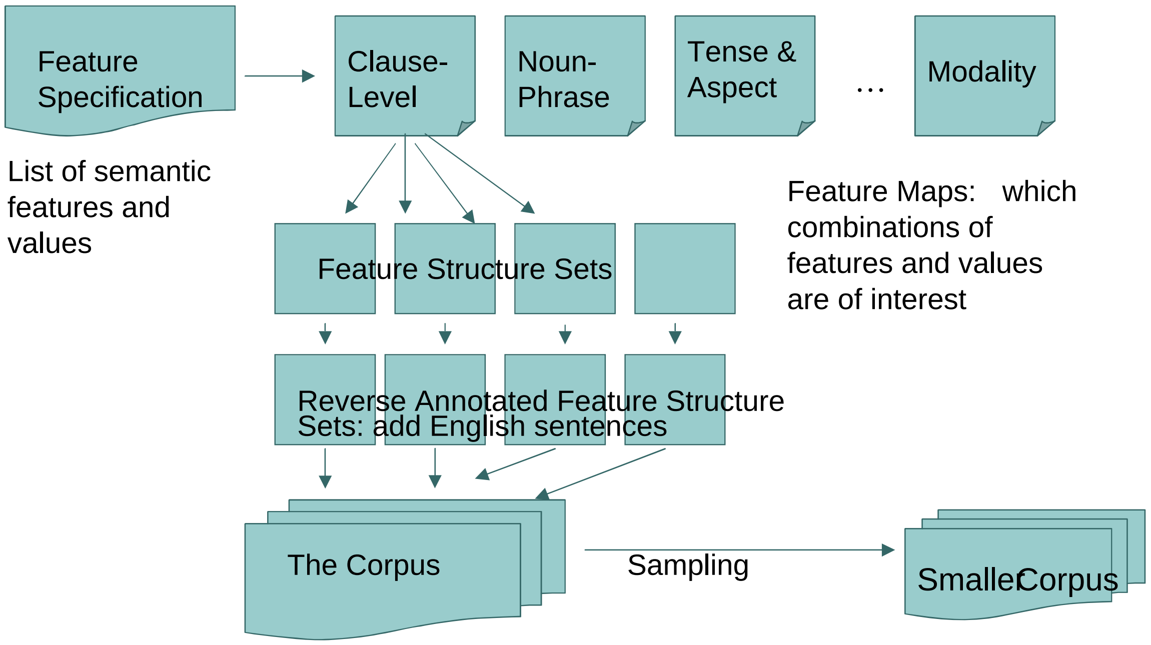

Feature Specification
Clause-Level
Noun-Phrase
Tense & Aspect
…
Modality
List of semantic features and values
Feature Maps: which combinations of features and values are of interest
Feature Structure Sets
Reverse Annotated Feature Structure Sets: add English sentences
The Corpus
Sampling
SmallerCorpus

Figure 3: An overview of the elicitation corpus production process

definiteness, tense, and modality, which are not in one-to-one correspondence with English words.

Creation of feature structures takes place in two steps. First, we define which combinations of features and values are of interest. Then the feature structures are automatically created from the feature specification.

Combinations of features are specified in Feature Maps (Figure 3). These maps identify features that are known to interact syntactically or morphologically in some languages. For example, tense in English is partially expressed using the auxiliary verb system. An unrelated aspect of meaning, whether a sentence is declarative or interrogative, interacts with the tense system in that it affects the word order of auxiliary verbs (*He was running*, *Was he running*), Thus there is an interaction of tense with interrogativity. We use studies of language typology to identify combinations of features that are known to interact.

Feature Maps are written in a concise formalism that is automatically expanded into a set of feature structures. For example, we can formally specify that we want three values of tense combined with three values of person, and nine feature structures will be produced. These are shown as Feature Structure Sets in Figure 3.

4 Sentence Writing

As stated previously, our corpus consists of feature structures that have been human annotated with a sentence and context field. Our feature structures contain functional-typological information, but do not contain specific lexical items. This means that our set of feature structures can be interpreted into any language using appropriate word choices and used for elicitation. Additionally, this leaves the human annotator with some freedom when selecting vocabulary items. Due to feedback from previous elicitation subjects we chose basic vocabulary words while steering clear of overly primitive subject matter that may be seen as insulting. Moreover, we did our best to avoid lexical gaps; for example, many languages do not have a single word that means *winner*.

Translator accuracy was also an important objective and we took pains to construct natural sounding, unambiguous sentences. The context field is used to clarify the sentence meaning and spell out features that may not manifest themselves in English.

5 Tools

In conjunction with this project we created several tools that can be reused to make new corpora with other purposes.

- An XML schema and XSLT can be used to make different feature specifications
- A feature structure generator that can be used as a guide to specify and design feature maps
- A feature structure browser can be used to make complicated feature structures easier to read and annotate

6 Conclusion

The basic steps for creating a functional-typological corpus are:

1. Combinations of features are selected
2. Sets of feature structures representing all feature combinations are generated
3. Humans write sentences with basic vocabulary that represent the meaning in the feature structure
4. If the corpus is too large, some or all of the corpus can be sampled

We used sampling and assessments of the most crucial features in order to compile our corpus and restrict it to a size small enough to be translatable by humans. As a result it is possible that this corpus will miss important feature combinations in some languages. However, a corpus containing all possible combinations of features would produce hundreds of billions of feature structures.

Our future research includes building a Corpus Navigation System to dynamically explore the full feature space. Using machine learning we will use information detected from translated sentences in order to decide what parts of the feature space are redundant and what parts must be explored and translated next. A further description of this process can be read in Levin et al. (2006).

Additionally, we will change from using humans to write sentences and context fields to having them generated by using a natural language generation system (Alvarez et al. 2005).

We also ran small scale experiments to measure translator accuracy and consistency and encountered positive results. Hebrew and Japanese translators provided consistent, accurate translations. Large scale experiments will be conducted in the near future to see if the success of the smaller experiments will carry over to a larger scale.

7 References

Alvarez, Alison, and Lori Levin, Robert Frederking, Jeff Good, Erik Peterson September 2005, Semi-Automated Elicitation Corpus Generation. In Proceedings of MT Summit X, Phuket: Thailand.

Bouquiaux, Luc and J.M.C. Thomas. 1992. *Studying and Describing Unwritten Languages.* Dallas, TX: The Summer Institute of Linguistcs.

Comrie, Bernard and N. Smith. 1977. *Lingua descriptive series: Questionnaire.* In: Lingua, 42:1-72.

Haspelmath, Martin and Matthew S. Dryer, David Gil, Bernard Comrie, editors. 2005 *World Atlas of Language Strucutures. Oxford University Press.*

Lori Levin, Alison Alvarez, Jeff Good, and Robert Frederking. 2006 "Automatic Learning of Grammatical Encoding." To appear in Jane Grimshaw, Joan Maling, Chris Manning, Joan Simpson and Annie Zaenen (eds) *Architectures, Rules and Preferences: A Festschrift for Joan Bresnan* , CSLI Publica tions. In Press.

Museli: A Multi-Source Evidence Integration Approach to Topic Segmentation of Spontaneous Dialogue

Jaime Arguello
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
jarguell@andrew.cmu.edu

Carolyn Rosé
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
cprose@cs.cmu.edu

Abstract

We introduce a novel topic segmentation approach that combines evidence of topic shifts from lexical cohesion with linguistic evidence such as syntactically distinct features of segment initial contributions. Our evaluation demonstrates that this hybrid approach outperforms state-of-the-art algorithms even when applied to loosely structured, spontaneous dialogue.

1 Introduction

Use of topic-based models of dialogue has played a role in information retrieval (Oard et al., 2004), information extraction (Baufaden, 2001), and summarization (Zechner, 2001). However, previous work on automatic topic segmentation has focused primarily on segmentation of expository text. We present Museli, a novel topic segmentation approach for dialogue that integrates evidence of topic shifts from lexical cohesion with linguistic indicators such as syntactically distinct features of segment initial contributions.

Our evaluation demonstrates that approaches designed for text do not generalize well to dialogue. We demonstrate a significant advantage of Museli over competing approaches. We then discuss why models based entirely on lexical cohesion fail on dialogue and how our algorithm compensates with other topic shift indicators.

2 Previous Work

Existing topic segmentation approaches can be loosely classified into two types: (1) lexical cohesion models, and (2) content-oriented models. The underlying assumption in lexical cohesion models is that a shift in term distribution signals a shift in topic (Halliday and Hassan, 1976). The best known algorithm based on this idea is TextTiling (Hearst, 1997). In TextTiling, a sliding window is passed over the vector-space representation of the text. At each position, the cosine correlation between the upper and lower region of the sliding window is compared with that of the peak cosine correlation values to the left and right of the window. A segment boundary is predicted when the magnitude of the difference exceeds a threshold.

One drawback to relying on term co-occurrence to signal topic continuity is that synonyms or related terms are treated as thematically-unrelated. One solution to this problem is using a dimensionality reduction technique such as Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997). Two such algorithms for segmentation are described in (Foltz, 1998) and (Olney and Cai, 2005).

Both TextTiling and Foltz's approach measure coherence as a function of the repetition of thematically-related terms. TextTiling looks for co-occurrences of terms or term-stems and Foltz uses LSA to measure semantic relatedness between terms. Olney and Cai's orthonormal basis approach also uses LSA, but allows a richer representation of discourse coherence, which is that coherence is a function of how much new information a discourse unit (e.g. a dialogue contribution) adds (*informativity)* and how relevant it is to the local context (*relevance*) (Olney and Cai, 2005).

Content-oriented models, such as (Barzilay and Lee, 2004), rely on the re-occurrence of patterns of topics over multiple realizations of thematically similar discourses, such as a series of newspaper articles about similar events. Their approach utilizes a hidden Markov model where states correspond to topics, and state transition probabilities correspond to topic shifts. To obtain the desired

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 9–12,
New York, June 2006. ©2006 Association for Computational Linguistics

number of topics (states), text spans of uniform length (individual contributions, in our case) are clustered. Then, state emission probabilities are induced using smoothed cluster-specific language models. Transition probabilities are induced by considering the proportion of documents in which a contribution assigned to the source cluster (state) immediately precedes a contribution assigned to the target cluster (state). Using an EM-like Viterbi approach, each contribution is reassigned to the state most likely to have generated it.

3 Overview of Museli Approach

We will demonstrate that lexical cohesion alone does not adequately mark topic boundaries in dialogue. Nevertheless, it can provide one meaningful source of evidence towards segmenting dialogue. In our hybrid Museli approach, we combined lexical cohesion with features that have the potential to capture something about the linguistic style that marks shifts in topic: word-unigrams, word-bigrams, and POS-bigrams for the current and previous contributions; the inclusion of at least one non-stopword term (contribution of content); time difference between contributions; contribution length; and the agent role of the previous and current contribution.

We cast the segmentation problem as a binary classification problem where each contribution is classified as NEW_TOPIC if the contribution introduces a new topic and SAME_TOPIC otherwise. We found that using a Naïve Bayes classifier (John & Langley, 1995) with an attribute selection wrapper using the chi-square test for ranking attributes performed better than other state-of-the-art machine learning algorithms, perhaps because of the evidence integration oriented nature of the problem. We conducted our evaluation using 10-fold cross-validation, being careful not to include instances from the same dialogue in both the training and test sets on any fold so that the results we report would not be biased by idiosyncratic communicative patterns associated with individual conversational participants picked up by the trained model.

Using the complete set of features enumerated above, we perform feature selection on the training data for each fold of the cross-validation separately, training a model with the top 1000 features, and applying that trained model to the test data. Examples of high ranking features confirm our intuition that contributions that begin new topic segments are syntactically marked. For example, many typical selected word bigrams were indicative of imperatives, such as lets-do, do-the, ok-lets, ok-try, lets-see, etc. Others included time oriented discourse markers such as now, then, next, etc.

To capitalize on differences in conversational behavior between participants assigned to different roles in the conversation (i.e., student and tutor in our evaluation corpora), we learn separate models for each role in the conversation[1]. This decision is based on the observation that participants with different agent-roles introduce topics with a different frequency, introduce different types of topics, and may introduce topics in a different style that displays their status in the conversation. For instance, a tutor may introduce new topics with a contribution that ends with an *imperative*. A student may introduce new topics with a contribution that ends with a *wh-question*.

4 Evaluation

In this section we evaluate Museli in comparison to the best performing state-of-the-art approaches, demonstrating that our hybrid Museli approach out-performs all of these approaches on two different dialogue corpora by a statistically significant margin ($p < .01$), in one case reducing the probability of error as measured by Beeferman's P_k to only 10% (Beeferman et al., 1999).

4.1 Experimental Corpora

We used two different dialogue corpora for our evaluation. The first corpus, which we refer to as the Olney & Cai corpus, is a set of dialogues selected randomly from the same corpus Olney and Cai selected their corpus from (Olney and Cai, 2005). The second corpus is a locally collected corpus of thermodynamics tutoring dialogues, which we refer to as the Thermo corpus. This corpus is particularly appropriate for addressing the research question of how to automatically segment dialogue for two reasons: First, the exploratory task that students and tutors engaged in together is more loosely structured than many task oriented domains typically investigated in the dialogue community, such as flight reservation or meeting scheduling. Second, because the tutor and student play asymmetric roles in the interaction, this corpus allows us to explore

[1] Dissimilar agent-roles occur in other domains as well (e.g. Travel Agent and Customer)

how conversational role affects how speakers mark topic shifts.

Table 1 presents statistics describing characteristics of these two corpora. Similar to (Passonneau and Litman, 1993), we adopt a flat model of topic-segmentation for our gold standard based on discourse segment purpose, where a shift in topic corresponds to a shift in purpose that is acknowledged and acted upon by both conversational agents. We evaluated inter-coder reliability over 10% of the Thermo corpus mentioned above. 3 annotators were given a 10 page coding manual with explanation of our informal definition of shared discourse segment purpose as well as examples of segmented dialogues. Pairwise inter-coder agreement was above 0.7 kappa for all pairs of annotators.

	Olney & Cai Corpus	**Thermo Corpus**
# Dialogues	42	22
Contributions/ Dialogue	195.40	217.90
Contributions/ Topic	24.00	13.31
Topics/Dialogue	8.14	16.36
Words/ Contribution	28.63	5.12

Table 1: Evaluation Corpora Statistics

4.2 Baseline Approaches

We evaluate Museli against the following algorithms: (1) Olney and Cai (Ortho), (2) Barzilay and Lee (B&L), (3) TextTiling (TT), and (4) Foltz.

As opposed to the other baseline algorithms, (Olney and Cai, 2005) applied their orthonormal basis approach specifically to dialogue, and prior to this work, report the highest numbers for topic segmentation of dialogue. Barzilay and Lee's approach is the state of the art in modeling topic shifts in monologue text. Our application of B&L to dialogue attempts to harness any existing and recognizable redundancy in topic-flow across our dialogues for the purpose of topic segmentation.

We chose TextTiling for its seminal contribution to monologue segmentation. TextTiling and Foltz consider lexical cohesion as their only evidence of topic shifts. Applying these approaches to dialogue segmentation sheds light on how term distribution in dialogue differs from that of expository monologue text (e.g. news articles).

The Foltz and Ortho approaches require a trained LSA space, which we prepared as described in (Olney and Cai, 2005). Any parameter tuning for approaches other than our hybrid approach was computed over the entire test set, giving competing algorithms the maximum advantage.

In addition to these approaches, we include segmentation results from three degenerate approaches: (1) classifying *all* contributions as NEW_TOPIC (ALL), (2) classifying *no* contributions as NEW_TOPIC (NONE), and (3) classifying contributions as NEW_TOPIC at *uniform intervals* (EVEN), corresponding to the average reference topic length (see Table 1).

As a means for comparison, we adopt two evaluation metrics: P_k and f-measure. An extensive argument of P_k's robustness (if k is set to ½ the average reference topic length) is present in (Beeferman, et al. 1999). P_k measures the probability of misclassifying two contributions a distance of k contributions apart, where the classification question is *are the two contributions part of the same topic segment or not?* Lower P_k values are preferred over higher ones. It equally captures the effect of false-negatives and false-positives and it favors near misses. F-measure punishes false positives equally, regardless of the distance to the reference boundary.

4.3 Results

Results for all approaches are displayed in Table 2. Note that lower values of P_k are preferred over higher ones. The opposite is true of F-measure. In both corpora, Museli performed significantly better than all other approaches ($p < .01$).

	Olney & Cai Corpus		**Thermo Corpus**	
	P_k	**F**	**P_k**	**F**
NONE	0.4897	--	0.4900	--
ALL	0.5180	--	0.5100	--
EVEN	0.5117	--	0.5132	--
TT	0.6240	0.1475	0.5353	0.1614
B&L	0.6351	0.1747	0.5086	0.1512
Foltz	0.3270	0.3492	0.5058	0.1180
Ortho	0.2754	0.6012	0.4898	0.2111
Museli	**0.1051**	**0.8013**	**0.4043**	**0.3693**

Table 2: Results on both corpora

4.4 Error Analysis

Results for all approaches are better on the Olney and Cai corpus than the Thermo corpus. The Thermo corpus differs profoundly from the Olney and Cai corpus in ways that very likely influenced the performance. For instance, in the *Thermo corpus* each dialogue contribution is an average of 5 words long, whereas in the *Olney and Cai corpus* each dialogue contribution contains an average of 28 words. Thus, the vector space representation of the dialogue contributions is much more sparse in the Thermo corpus, which makes shifts in lexical coherence less reliable as topic shift indicators.

In terms of P_k, TextTiling (TT) performed worse than the degenerate algorithms. TextTiling measures the term-overlap between adjacent regions in the discourse. However, dialogue contributions are often terse or even contentless. This produces many islands of contribution-sequences for which the local lexical cohesion is zero. TextTiling wrongfully classifies all of these as starts of new topics. A heuristic improvement to prevent TextTiling from placing topic boundaries at every point along a sequence of contributions failed to produce a statistically significant improvement.

The Foltz and the orthonormal basis approaches rely on LSA to provide strategic semantic generalizations. Following (Olney and Cai, 2005), we built our LSA space using dialogue contributions as the atomic text unit. However, in corpora such as the Thermo corpus, this may not be effective because of the brevity of contributions.

Barzilay and Lee's algorithm (B&L) did not generalize well to either dialogue corpus. One reason could be that such probabilistic methods require that reference topics have significantly different language models, which was not true in either of our evaluation corpora. We also noticed a number of instances in the dialogue corpora where participants referred to information from previous topic segments, which consequently may have blurred the distinction between the language models assigned to different topics.

5 Current Directions

In this paper we address the problem of automatic topic segmentation of spontaneous dialogue. We demonstrated with an empirical evaluation that state-of-the-art approaches fail on spontaneous dialogue because word-distribution patterns alone are insufficient evidence of topic shifts in dialogue. We have presented a supervised learning algorithm for topic segmentation of dialogue that combines linguistic features signaling a contribution's function with lexical cohesion. Our evaluation on two distinct dialogue corpora shows a significant improvement over the state of the art approaches.

The disadvantage of our approach is that it requires hand-labeled training data. We are currently exploring ways of bootstrapping a model from a small amount of hand labeled data in combination with lexical cohesion (tuned for high precision and consequently low recall) and some reliable discourse markers.

Acknowledgments

This work was funded by Office of Naval Research, Cognitive and Neural Science Division, grant number N00014-05-1-0043.

References

Regina Barzilay and Lillian Lee (2004). Catching the drift: Probabilistic Content Models, with Applications to Generation and Summarization. In *Proceedings of HLT-NAACL 2004.*

Doug Beeferman, Adam Berger, John D. Lafferty (1999). Statistical Models for Text Segmentation. *Machine Learning* 34 (1-3): 177-210.

Narjès Boufaden, Guy Lapalme, Yoshua Bengio (2001). Topic Segmentation: A first stage to Dialog-based Information Extraction. In *Proceedings of NLPRS 2001*.

P.W. Foltz, W. Kintsch, and Thomas Landauer (1998). The measurement of textual cohesion with latent semantic analysis. *Discourse Processes*, 25, 285-307.

M. A. K. Halliday and Ruqaiya Hasan (1976). *Cohesion in English*. London: Longman.

Marti Hearst. 1997. TextTiling: Segmenting Text into Multi-Paragragh Subtopic Passages. *Computational Linguistics*, 23(1), 33 – 64.

George John & Pat Langley (1995). Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of UAI* 2005.

Thomas Landauer, & Susan Dumais (1997). A Solution to Plato's Problem: The Latent Semantic Analysis of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104, 221-240.

Douglas Oard, Bhuvana Ramabhadran, and Samuel Gustman (2004). Building an Information Retrieval Test Collection for Spontaneous Conversational Speech. In *Proceedings of SIGIR* 2004.

Andrew Olney and Zhiqiang Cai (2005). An Orthonormal Basis for Topic Segmentation of Tutorial Dialogue. In *Proceedings of HLT-EMNLP* 2005.

Rebecca Passonneau and Diane Litman (1993). Intention-Based Segmentation: Human Reliability and Correlation with Linguistic Cues. In *Proceedings ACL* 2003.

Klaus Zechner (2001). Automatic Generation of Concise Summaries of Spoken Dialogues in Unrestricted Domains. In *Proceedings of SIGIR 2001*.

Measuring Semantic Relatedness Using People and WordNet

Beata Beigman Klebanov
School of Computer Science and Engineering
The Hebrew University, Jerusalem, Israel
beata@cs.huji.ac.il

Abstract

In this paper, we (1) propose a new dataset for testing the degree of relatedness between pairs of words; (2) propose a new WordNet-based measure of relatedness, and evaluate it on the new dataset.

1 Introduction

Estimating the degree of semantic relatedness between words in a text is deemed important in numerous applications: word-sense disambiguation (Banerjee and Pedersen, 2003), story segmentation (Stokes et al., 2004), error correction (Hirst and Budanitsky, 2005), summarization (Barzilay and Elhadad, 1997; Gurevych and Strube, 2004).

Furthermore, Budanitsky and Hirst (2006) noted that various applications tend to pick the same measures of relatedness, which suggests a certain commonality in what is required from such a measure by the different applications. It thus seems worthwhile to develop such measures intrinsically, before putting them to application-based utility tests.

The most popular, by-now-standard testbed is Rubenstein and Goodenough's (1965) list of 65 noun pairs, ranked by similarity of meaning. A 30-pair subset (henceforth, **MC**) passed a number of replications (Miller and Charles, 1991; Resnik, 1995), and is thus highly reliable.

Rubenstein and Goodenough (1965) view similarity of meaning as degree of synonymy. Researchers have long recognized, however, that synonymy is only one kind of semantic affinity between words in a text (Halliday and Hasan, 1976), and expressed a wish for a dataset for testing a more general notion of semantic relatedness.[1]

This paper proposes and explores a new relatedness dataset. In sections 2-3, we briefly introduce the experiment by Beigman Klebanov and Shamir (henceforth, **BS**), and use the data to induce relatedness scores. In section 4, we propose a new WordNet-based measure of relatedness, and use it to explore the new dataset. We show that it usually does better than competing WordNet-based measures (section 5). We discuss future directions in section 6.

2 Data

Aiming at reader-based exploration of lexical cohesion in texts, Beigman Klebanov and Shamir conducted an experiment with 22 students, each reading 10 texts: 3 news stories, 4 journalistic and 3 fiction pieces (Beigman Klebanov and Shamir, 2006). People were instructed to read the text first, and then go over a separately attached list of words in order of their appearance in the text, and ask themselves, for every newly mentioned concept, "which previously mentioned concepts help the easy accommodation of the current concept into the evolving story, if indeed it is easily accommodated, based on the common knowledge as perceived by the annotator" (Beigman Klebanov and Shamir, 2005); this preceding helper concept is called an *anchor*. People were asked to mark all anchoring relations they could find.

The rendering of relatedness between two concepts is not tied to any specific lexical relation, but rather to common-sense knowledge, which has to do with "knowledge of kinds, of associations, of typical situations, and even typical utterances".[2] The phenomenon is thus clearly construed as much broader than degree-of-synonymy.

Beigman Klebanov and Shamir (2006) provide reliability estimation of the experimental data using

[1] "...similarity of meaning is not the same thing as semantic relatedness. However, there is at present no large dataset of human judgments of semantic relatedness" (Hirst and Budanitsky, 2005); "To our knowledge, no datasets are available for validating the results of semantic relatedness metric" (Gurevych, 2005).

[2] according to Hirst (2000), cited in the guidelines

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 13–16,
New York, June 2006. ©2006 Association for Computational Linguistics

statistical analysis and a validation experiment, identifying reliably anchored items with their strong anchors, and reliably un-anchored items. Such analysis provides high-validity data for classification; however, much of the data regarding intermediate degrees of relatedness is left out.

3 Relatedness Scores

Our idea is to induce scores for pairs of anchored items with their anchors (henceforth, **AApairs**) using the cumulative annotations by 20 people.[3] Thus, an AApair written by all 20 people scores 20, and that written by just one person scores 1. The scores would correspond to the perceived relatedness of the pair of concepts in the given text.

In Beigman Klebanov and Shamir's (2006) core classification data, no distinctions are retained between pairs marked by 19 or 13 people. Now we are interested in the relative relatedness, so it is important to handle cases where the BS data might under-rate a pair. One such case are multi-word items; we remove AApairs with suspect multi-word elements.[4] Further, we retain only pairs that belong to open-class parts of speech (henceforth, **POS**), as functional categories contribute little to the lexical texture (Halliday and Hasan, 1976). The *Size* column of table 1 shows the number of AApairs for each BS text, after the aforementioned exclusions.

The induced scores correspond to cumulative judgements of a group of people. How well do they represent the people's ideas? One way to measure group homogeneity is leave-one-out estimation, as done by Resnik (1995) for MC data, attaining the high average correlation of $r = 0.88$. In the current case, however, every specific person made a binary decision, whereas a group is represented by scores 1 to 20; such difference in granularity is problematic for correlation or rank order analysis.

Another way to measure group homogeneity is to split it into subgroups and compare scores emerging from the different subgroups. We know from Beigman Klebanov and Shamir's (2006) analysis that it is not the case that the 20-subject group clusters into subgroups that systematically produced different patterns of answers. This leads us to expect relative lack of sensitivity to the exact splits into subgroups.

To validate this reasoning, we performed 100 random choices of two 9-subject[4] groups, calculated the scores induced by the two groups, and computed Pearson correlation between the two lists. Thus, for every BS text, we have a distribution of 100 coefficients, which is approximately normal. Estimations of μ and σ of these distributions are $\mu = .69 - .82$ (av. 0.75), $\sigma = .02 - .03$ for the different BS texts.

To summarize: although the homogeneity is lower than for MC data, we observe good average intergroup correlations with little deviation across the 100 splits. We now turn to discussion of a relatedness measure, which we will evaluate using the data.

4 Gic: WordNet-based Measure

Measures using WordNet taxonomy are state-of-the-art in capturing semantic similarity, attaining r=.85 −.89 correlations with the MC dataset (Jiang and Conrath, 1997; Budanitsky and Hirst, 2006). However, they fall short of measuring relatedness, as, operating within a single-POS taxonomy, they cannot meaningfully compare *kill* to *death*. This is a major limitation with respect to BS data, where only about 40% of pairs are nominal, and less than 10% are verbal. We develop a WordNet-based measure that would allow cross-POS comparisons, using glosses in addition to the taxonomy.

One family of WordNet measures are methods based on estimation of information content (henceforth, **IC**) of concepts, as proposed in (Resnik, 1995). Resnik's key idea in corpus-based information content induction using a taxonomy is to count every appearance of a concept as mentions of all its hypernyms as well. This way, *artifact#n#1*, although rarely mentioned explicitly, receives high frequency and low IC value. We will count a concept's mention towards all its hypernyms AND all words[5] that appear in its own and its hypernyms' glosses. Analogously to *artifact*, we expect properties mentioned in glosses of more general concepts to be less informative, as those pertain to more things (ex., *visible*, a property of anything that is-a *physical object*). The details of the algorithm for information content induction from taxonomy and gloss information (IC_{GT}) are given in appendix A.

To estimate the semantic affinity between two senses A and B, we average the IC$_{GT}$ values of the 3 words with the highest IC$_{GT}$ in the overlap of A's and B's expanded glosses (the expansion follows the algorithm in appendix A).[6]

[3] Two subjects were revealed as outliers and their data was removed (Beigman Klebanov and Shamir, 2006).

[4] See Beigman Klebanov (2006) for details.

[5] We induce IC values on (POS-tagged base form) words rather than senses. Ongoing gloss sense-tagging projects like eXtended WordNet (http://xwn.hlt.utdallas.edu/links.html) would allow sense-based calculation in the future.

[6] The number 3 is empirically-based; the idea is to counter-balance (a) the effect of an accidental match of a

Data	Size	Gic	BP	Data	Size	Gic	BP
BS-1	1007	.29	.19	BS-6	536	.24	.19
BS-2	776	.37	.16	BS-7	917	.22	.10
BS-3	1015	.22	.09	BS-8	529	.24	.12
BS-4	512	.34	.39	BS-9	509	.31	.16
BS-5	1020	.25	.11	BS10	417	.36	.19

Table 1: Dataset sizes and correlations of Gic, BP with human ratings. $r > 0.16$ is significant at $p < .05$; $r > .23$ is significant at $p < .01$. Average correlation (Av_{BS}) is r=.28 (Gic), r=.17 (BP).

If A^* (the word of which A is a sense) appears in the expanded gloss of B, we take the maximum between the $\mathrm{IC}_{GT}(A^*)$ and the value returned by the 3-smoothed calculation. To compare two words, we take the maximum value returned by pairwise comparisons of their WordNet senses.[7]

The performance of this measure is shown under **Gic** in table 1. Gic manages robust but weak correlations, never reaching the $r = .40$ threshold.

5 Related Work

We compare Gic to another WordNet-based measure that can handle cross-POS comparisons, proposed by Banerjee and Pedersen (2003). To compare word senses A and B, the algorithm compares not only their glosses, but also glosses of items standing in various WordNet relations with A and B. For example, it compares the gloss of A's meronym to that of B's hyponym. We use the default configuration of the measure in WordNet::Similarity-0.12 package (Pedersen et al., 2004), and, with a single exception, the measure performed below Gic; see **BP** in table 1.

As mentioned before, taxonomy-based *similarity* measures cannot fully handle BS data. Table 2 uses nominal-only subsets of BS data and the MC nominal similarity dataset to show that (a) state-of-the-art WordNet-based similarity measure **JC**[8] (Jiang and Conrath, 1997; Budanitsky and Hirst, 2006) does very poorly on the relatedness data, suggesting that nominal similarity and relatedness are rather different things; (b) Gic does better on average, and is more robust; (c) Gic yields on MC to gain performance on BS, whereas BP is no more inclined towards relatedness than JC.

Data	Gic	BP	JC	Data	Gic	BP	JC
BS-1	.38	.18	.21	BS-6	.25	.16	.22
BS-2	.53	.18	.37	BS-7	.23	.10	.04
BS-3	.21	.04	.01	BS-8	.32	.10	.00
BS-4	.28	.38	.33	BS-9	.24	.17	.27
BS-5	.12	.07	.16	BS10	.41	.25	.25
Av_{BS}	.30	.16	.19	MC	.78	.80	.86

Table 2: MC and nominal-only subsets of BS: correlations of various measures with the human ratings.

Table 3 illustrates the relatedness vs. similarity distinction. Whereas, taxonomically speaking, *son* is more similar to *man*, as reflected in JC scores, people marked *family* and *mother* as much stronger anchors for *son* in BS-2; Gic follows suit.

AApair	Human	Gic	JC
son – man	2	0.355	22.3
son – family	13	0.375	16.9
son – mother	16	0.370	20.1

Table 3: Relatendess vs. similarity

6 Conclusion and Future Work

We proposed a dataset of relatedness judgements that differs from the existing ones[9] in (1) size – about 7000 items, as opposed to up to 350 in existing datasets; (2) cross-POS data, as opposed to purely nominal or verbal; (3) a broad approach to semantic relatedness, not focussing on any particular relation, but grounding it in the reader's (idea of) common knowledge; this as opposed to synonymy-based similarity prevalent in existing databases.

We explored the new data with WordNet-based measures, showing that (1) the data is different in character from a standard similarity dataset, and very challenging for state-of-the-art methods; (2) the proposed novel WordNet-based measure of relatedness usually outperforms its competitor, as well as a state-of-the-art similarity measure when the latter applies.

In future work, we plan to explore distributional methods for modeling relatedness, as well as the use of text-based information to improve correlations with the human data, as judgments are situated in specific textual contexts.

single word which is relatively rarely used in glosses; (b) the multitude of low-IC items in many of the overlaps that tend to downplay the impact of the few higher-IC members of the overlap.

[7] To speed the processing up, we use first 5 WordNet senses of each item for results reported here.

[8] See formula in appendix B. We use (Pedersen et al., 2004) implementation with a minor alteration – see Beigman Klebanov (2006).

[9] Though most widely used, MC is not the only available dataset; we will address other datasets in a subsequent paper.

References

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of IJCAI*.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of ACL Intelligent Scalable Text Summarization Workshop*.

Beata Beigman Klebanov and Eli Shamir. 2005. Guidelines for annotation of concept mention patterns. Technical Report 2005-8, Leibniz Center for Research in Computer Science, The Hebrew University of Jerusalem, Israel.

Beata Beigman Klebanov and Eli Shamir. 2006. Reader-based exploration of lexical cohesion. *To appear in Language Resources and Evaluation*. Springer, Netherlands.

Beata Beigman Klebanov. 2006. Using people and WordNet to measure semantic relatedness. Technical Report 2006-17, Leibniz Center for Research in Computer Science, The Hebrew University of Jerusalem, Israel.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47.

Iryna Gurevych and Michael Strube. 2004. Semantic similarity applied to spoken dialogue summarization. In *Proceedings of COLING*.

Iryna Gurevych. 2005. Using the structure of a conceptual network in computing semantic relatedness. In *Proceedings of IJCNLP*.

M.A.K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman Group Ltd.

Graeme Hirst and Alexander Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1):87–111.

Graeme Hirst. 2000. Context as a spurious concept. In *Proceedings of CICLING*.

Jay Jiang and David Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings on International Conference on Research in Computational Linguistics*.

George Miller and Walter Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity-measuring the relatedness of concepts. In *Proceedings of NAACL*.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI*.

Herbert Rubenstein and John Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Nicola Stokes, Joe Carthy, and Alan F. Smeaton. 2004. SeLeCT: A lexical cohesion based news story segmentation system. *Journal of AI Communications*, 17(1):3–12.

A Gloss&Taxonomy IC (IC_{GT})

We refer to POS-tagged base form items as "words" throughout this section. For every word-sense W in WordNet database for a given POS:

1. Collect all content words from the gloss of W, excluding examples, including W^* - the POS-tagged word of which W is a sense.
2. If W is part of a taxonomy, expand its gloss, without repetitions, with words appearing in the glosses of all its super-ordinate concepts, up to the top of the hierarchy. Thus, the expanded gloss for *airplane#n#1* would contain words from the glosses of the relevant senses of *aircraft* , *vehicle*, *transport*, etc.
3. Add W's sense count to all words in its expanded gloss.[10]

Each POS database induces its own counts on each word that appeared in the gloss of at least one of its members. When merging the data from the different POS, we scale the aggregated counts, such that they correspond to the proportion of the given word in the POS database where it was the least informative. The standard log-frequency calculation transforms these counts into taxonomy-and-gloss based information content (IC_{GT}) values.

B JC measure of similarity

In the formula, IC is taxonomy-only based information content, as in (Resnik, 1995), LS is the lowest common subsumer of the two concepts in the WordNet hierarchy, and Max is the maximum distance[11] between any two concepts.

$$JC(c_1, c_2) = Max-(IC(c_1)+IC(c_2)-2\times IC(LS(c_1, c_2))$$

To make JC scores comparable to Gic's [0,1] range, the score can be divided by Max. Normalization has no effect on correlations.

[10] We do add-1-smoothing on WordNet sense counts.

[11] This is about 26 for WordNet-2.0 nominal hierarchy with add-1-smoothed SemCor database; see Beigman Klebanov (2006) for details.

Thai Grapheme-Based Speech Recognition

Paisarn Charoenpornsawat, Sanjika Hewavitharana, Tanja Schultz

Interactive Systems Laboratories, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{paisarn, sanjika, tanja}@cs.cmu.edu

Abstract

In this paper we present the results for building a grapheme-based speech recognition system for Thai. We experiment with different settings for the initial context independent system, different number of acoustic models and different contexts for the speech unit. In addition, we investigate the potential of an enhanced tree clustering method as a way of sharing parameters across models. We compare our system with two phoneme-based systems; one that uses a hand-crafted dictionary and another that uses an automatically generated dictionary. Experiment results show that the grapheme-based system with enhanced tree clustering outperforms the phoneme-based system using an automatically generated dictionary, and has comparable results to the phoneme-based system with the hand-crafted dictionary.

1 Introduction

Large vocabulary speech recognition systems traditionally use phonemes as sub-word units. This requires a pronunciation dictionary, which maps the orthographic representation of words into a sequence of phonemes. The generation of such a dictionary is both time consuming and expensive since it often requires linguistic knowledge of the target language. Several approaches to automatic dictionary generation have been introduced in the past with varying degrees of success (Besling, 1994; Black et al., 1998). Nevertheless, these methods still require post editing by a human expert or using another manually generated pronunciation dictionary.

As a solution to this problem, grapheme-based speech recognition (GBSR) has been proposed recently (Kanthak and Ney, 2002). Here, instead of phonemes, graphemes – orthographic representation of a word – are used as the sub word units. This makes the generation of the pronunciation dictionary a trivial task. GBSR systems have been successfully applied to several European languages (Killer et al., 2003). However, because of the generally looser relation of graphemes to pronunciation than phonemes, the use of context dependent modeling techniques and the sharing of parameters across different models are of central importance.

The variations in the pronunciation of phonemes in different contexts are usually handled by clustering the similar contexts together. In the traditional approach, decision trees are used to cluster polyphones – a phoneme in a specific context – together. Due to computational and memory constraints, individual trees are grown for each sub-state of each phoneme. This does not allow the sharing of parameters across polyphones with different center phonemes. Enhanced tree clustering (Yu and Schultz, 2003) lifts this constraint by growing trees which cover multiple phonemes.

In this paper we present our experiments on applying grapheme-based speech recognition for Thai language. We compare the performance of the grapheme-based system with two phoneme-based systems, one using a hand-crafter dictionary, and the other using an automatically generated diction-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 17–20,
New York, June 2006. ©2006 Association for Computational Linguistics

ary. In addition, we observe the effect of the enhanced tree clustering on the grapheme-based recognition system.

2 Grapheme-to-Phoneme Relation in Thai

In the grapheme-based approach, the pronunciation dictionary is constructed by splitting a word into its constituent letters. Previous experiments have shown that the quality of the grapheme-based recognizer is highly dependent on the nature of the grapheme-to-phoneme relation of a specific language (Killer, 2003). In this section we have a closer look at the grapheme-to-phoneme relation in Thai.

Thai, an alphabetical language, has 44 letters for 21 consonant sounds, 19 letters for 24 vowel sounds (9 short vowels, 9 long vowels and 6 diphthongs), 4 letters for tone markers (5 tones), few special letters, and numerals. There are some characteristics of Thai writing that can cause problems for GBSR:

- Some vowel letters can appear before, after, above or below a consonant letter. e.g. In the word "**แมว**" (/mae:w/), the vowel "**แ**" (/ae:/) appears before the consonant "**ม**" (/m/).
- Some vowel and consonant letters can be combined together to make a new vowel. e.g. In the word "**มัว**" /mua/, the vowel "ua" is composed of a vowel letter " ั " and a consonant letter "**ว**".
- Some vowels are represented by more than one vowel letter For example, the vowel /ae/ requires two vowel letters: "**แ**" and "**ะ**". To make a syllable, a consonant is inserted in between the two vowel letters. e.g. "**และ**" (/lae/). The consonant "**ล**" (/l/) is in the middle.
- In some syllables, vowels letters are not explicitly written. e.g. The word "**ยก**" (/yok/) consists of two consonant letter, "**ย**" (/y/) and "**ก**" (/k/). There is no letter to represent the vowel /o/.
- The special letter " ์ ", called Karan, is a deletion marker. If it appears above a consonant, that consonant will be ignored. Sometimes, it can also delete the immediately preceding consonant or the whole syllable.

To make the relationship between graphemes and phonemes in Thai as close as possible we apply two preprocess steps:

- Reordering of graphemes when a vowel comes before a consonant.
- Merging multiple letters representing a single phoneme into one symbol.

We use simple heuristic rules for this purpose; 10 rules for reordering and 15 for merging. In our initial experiments, reordering alone gave better results than reordering plus merging. Hence, we only used reordering rules for the rest of the experiments.

3 Thai Grapheme-Based Speech Recognition

In this section, we explain the details of our Thai GBSR system. We used the Thai GlobalPhone corpus (Suebvisai et.al., 2005) as our data set, which consists of read-speech in the news domain. The corpus contains 20 hours of recorded speech from 90 native Thai speakers consisting of 14k utterances. There are approximately 260k words covering a vocabulary of about 7,400 words. For testing we used 1,181 utterances from 8 different speakers. The rest was used for training. The language model was built on news articles and gave a trigram perplexity of 140 and an OOV-rate of 1.4% on the test set.

To start building the acoustic models for Thai, we first used a distribution that equally divided the number of frames among the graphemes. This was then trained for six iterations followed by writing the new labels. We repeated these steps six times. As can be seen in Table 1, the resulting system (Flat-Start) had poor performance. Hence we decided to bootstrap from a context independent acoustic model of an existing phoneme-based speech recognition (PBSR) systems.

3.1 Bootstrapping

We trained two grapheme-based systems by bootstrapping from the acoustic models of two different PBSR systems. The first system (Thai) was bootstrapped from a Thai PBSR system (Suebvisai et al., 2005) trained on the same corpus. The second system (Multilingual) was bootstrapped from the acoustic models trained on the multilingual GlobalPhone corpus (Schultz and Waibel, 1998) which shares acoustic models of similar sounds across multiple languages. In mapping phones to graphemes, when a grapheme can be mapped to several different phones we selected the one which occurs more frequently.

Both systems were based on trigraphemes (+/-1) with 500 acoustic models. Training was identical to the Flat-Start system. Table 1 compares the word error rates (WER) of the three systems on the test set.

Flat-Start	Multilingual	Thai
37.2%	27.0 %	26.4 %

Table 1: Word error rates in % of GBSR systems with different bootstrapping techniques

Results show that the two bootstrapped systems have comparable results, while Thai system gives the lowest WER. For the rest of the experiments we used the system bootstrapped from the multilingual acoustic models.

3.2 Building Context Dependent Systems

For the context dependent systems, we trained two systems each with different polygrapheme units; one with trigrapheme (+/- 1), and another with quintgrapheme (+/-2).

The question set used in building the context dependent system was manually constructed by using the question set from the Thai PBSR system. Then we replaced every phoneme in the question set by the appropriate grapheme(s). In addition, we compared two different acoustic model sizes; 500 and 2000 acoustic models.

Table 2 shows the recognition results for the resulting GBSR systems.

Speech Unit	500 models	2000 models
Trigrapheme	26.0 %	26.0 %
Quintgrapheme	27.0 %	30.3 %

Table 2: Word error rates in % of GBSR systems using different speech units and the # of models.

The system with 500 acoustic models based on trigraphemes produced the best results. The higher WER for the quintgrapheme system might be due to the data sparseness.

3.3 Enhanced Tree Clustering (ETC)

Yu and Schultz (2003) introduced a tree clustering approach that allows the sharing of parameters across phonemes. In this enhanced tree clustering, a single decision tree is constructed for all sub-states of all phonemes. The clustering procedure starts with all the polyphones at the root of the tree. The decision tree can ask questions regarding the identity of the center phoneme and its neighboring phonemes, plus the sub-state identity (begin/middle/end). At each node, the question that yields the highest information gain is chosen and the tree is split. This process is repeated until the tree reaches a certain size. Enhanced tree clustering is well suited to implicitly capture the pronunciation variations in speech by allowing certain polyphones that are pronounced similarly to share the same set of parameters. Mimer et al. (2004) shows that this approach can successfully be applied to grapheme based speech recognition by building separate trees for each sub-state for consonants and vowels.

For the experiments on enhanced tree clustering, we used the same setting as the grapheme-based system. Instead of growing a single tree, we built six separate trees – one each for begin, middle and end sub-states of vowels and consonants. Apart from the question set used in the grapheme-based system, we added singleton questions, which ask about the identity of different graphemes in a certain context. To apply the decision tree algorithm, a semi-continuous recognition system was trained. Since the number of models that share the same codebook drastically increases, we increased the number of Gaussians per codebook. Two different values were tested; 500 (ETC-500) and 1500 (ETC-1500) Gaussians. Table 4 shows the recognition results on the test set, after applying enhanced tree clustering to the system based on trigraphemes (MUL-TRI).

	500 models	2000 models
MUL-TRI	26.0 %	26.0 %
ETC-500	16.9 %	18.0 %
ETC-1500	18.1 %	19.0 %

Table 3: Word error rate in % for the enhance tree clustering method

As can be seen from Table 3, the enhanced tree clustering has significant improvement over the best grapheme-based system. ETC-500 with relatively lesser number of parameters has outperformed ETC-1500 system. Performance decreases when we increase the number of leaf nodes in the tree, from 500 to 2000. A closer look at the cluster trees that used the enhanced clustering reveals that

50~100 models share parameters across different center graphemes.

4 Grapheme vs. Phoneme based SR

To evaluate our grapheme-based approach with the traditional phoneme-based approach, we compared the best GBSR system with two phoneme-based systems.

The first system (PB-Man) uses a manually created dictionary and is identical to (Suebvisai et al., 2005) except that we used triphones as the speech unit. The second system (PB-LTS) uses an automatically generated dictionary using letter-to-sound rules. To generate the dictionary in PB-LTS, we used the letter-to-sound rules in Festival (Black 1998) speech synthesis system trained with 20k words. We also applied the same reordering rules used in the GBSR system as described in section 2. Both the systems have 500 acoustic models based on triphones.

Table 4 gives the WER for the two systems, on the test set. Best results from GBSR systems are also reproduced here for the comparison.

Phoneme-based	
Using manual dictionary (PB-Man)	16.0 %
Using automatic dictionary (PB-LTS)	24.5%
Grapheme-based	
MUL-TRI	26.0 %
MUL-TRI with ETC (ETC-500)	16.9 %

Table 4: Word error rates in % of GBSR and PBSR systems

As expected, the manually generated dictionary gives the best performance. The performance between PB-LTS and grapheme based system are comparable. ETC-500 system has a significantly better performance than the automatically generated dictionary, and almost the same results as the phoneme-based baseline. This shows that grapheme-based speech recognition coupled with the enhanced tree clustering can be successfully applied to Thai speech recognition without the need for a manually generated dictionary.

5 Conclusions

In this paper we presented the results for applying grapheme-based speech recognition to Thai language. We experimented with different settings for the initial context independent system, different number of acoustic models and different contexts for the polygraphemes. We also tried the enhanced tree clustering method as a means of sharing parameters across models. The results show that the system with 500 acoustic models based on trigraphemes produce the best results. Additionally, the enhanced tree clustering significantly improves the recognition accuracy of the grapheme-based system. Our system outperformed a phoneme-based system that uses an automatically generated dictionary. These results are very promising since they show that the grapheme-based approach can be successfully used to generate speech recognition systems for new languages using little linguistic knowledge.

References

Stefan Besling. 1994. “Heuristical and Statistical Methods for Grapheme-to-Phoneme Conversion. In Proceedings of Konvens. Vienna, Austria.

Alan W. Black, Kevin Lenzo and Vincent Pagel. 1998. Issues in Building General Letter to Sound Rules. In *Proceedings of the ESCA Workshop on Speech Synthesis*, Australia.

Sebastian Kanthak and Hermann Ney. 2002. Context-dependent Acoustic Modeling using Graphemes for Large Vocabulary Speech Recognition. In *Proceedings of the ICASSP*. Orlando, Florida.

Mirjam Killer, Sebastian Stüker, and Tanja Schultz. 2003. Grapheme Based Speech Recognition. In *Proceeding of the Eurospeech*. Geneva, Switzerland.

Borislava Mimer, Sebastian Stüker, and Tanja Schultz. 2004. Flexible Decision Trees for Grapheme Based Speech Recognition. In *Proceedings of the 15th Conference Elektronische Sprachsignalverarbeitung (ESSV)*, Cotbus, Germany, September.

Tanja Schultz and Alex Waibel. 1998. Development of Multi-lingual Acoustic Models in the GlobalPhone Project. In *Proceedings of the 1st Workshop on Text, Speech, and Dialogue (TSD)*, Brno, Czech Republic.

Sinaporn Suebvisai, Paisarn Charoenpornsawat, Alan Black and et.al. 2005 Thai Automatic Speech Recognition. Proceedings of ICASSP, Philadelphia, Pennsylvania.

Hua Yu and Tanja Schultz. 2003. Enhanced Tree Clustering with Single Pronunciation dictionary for Conversational Speech Recognition. In *Proceedings of the 8th Eurospeech*, Geneva, Switzerland.

Class Model Adaptation for Speech Summarisation

Pierre Chatain, Edward W.D. Whittaker, Joanna Mrozinski and Sadaoki Furui
Dept. of Computer Science
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
{pierre, edw, mrozinsk, furui}@furui.cs.titech.ac.jp

Abstract

The performance of automatic speech summarisation has been improved in previous experiments by using linguistic model adaptation. We extend such adaptation to the use of class models, whose robustness further improves summarisation performance on a wider variety of objective evaluation metrics such as ROUGE-2 and ROUGE-SU4 used in the text summarisation literature. Summaries made from automatic speech recogniser transcriptions benefit from relative improvements ranging from 6.0% to 22.2% on all investigated metrics.

1 Introduction

Techniques for automatically summarising written text have been actively investigated in the field of natural language processing, and more recently new techniques have been developed for speech summarisation (Kikuchi et al., 2003). However it is still very hard to obtain good quality summaries. Moreover, recognition accuracy is still around 30% on spontaneous speech tasks, in contrast to speech read from text such as broadcast news. Spontaneous speech is characterised by disfluencies, repetitions, repairs, and fillers, all of which make recognition and consequently speech summarisation more difficult (Zechner, 2002). In a previous study (Chatain et al., 2006), linguistic model (LiM) adaptation using different types of word models has proved useful in order to improve summary quality. However sparsity of the data available for adaptation makes it difficult to obtain reliable estimates of word n-gram probabilities. In speech recognition, class models are often used in such cases to improve model robustness. In this paper we extend the work previously done on adapting the linguistic model of the speech summariser by investigating class models. We also use a wider variety of objective evaluation metrics to corroborate results.

2 Summarisation Method

The summarisation system used in this paper is essentially the same as the one described in (Kikuchi et al., 2003), which involves a two step summarisation process, consisting of sentence extraction and sentence compaction. Practically, only the sentence extraction part was used in this paper, as preliminary experiments showed that compaction had little impact on results for the data used in this study.

Important sentences are first extracted according to the following score for each sentence $W = w_1, w_2, ..., w_n$, obtained from the automatic speech recognition output:

$$S(W) = \frac{1}{N}\sum_{i=1}^{N}\{\alpha_C C(w_i) + \alpha_I I(w_i) + \alpha_L L(w_i)\}, \tag{1}$$

where N is the number of words in the sentence W, and $C(w_i)$, $I(w_i)$ and $L(w_i)$ are the confidence score, the significance score and the linguistic score of word w_i, respectively. α_C, α_I and α_L are the respective weighting factors of those scores, determined experimentally.

For each word from the automatic speech recogni-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 21–24,
New York, June 2006.

tion transcription, a logarithmic value of its posterior probability, the ratio of a word hypothesis probability to that of all other hypotheses, is calculated using a word graph obtained from the speech recogniser and used as a confidence score.

For the significance score, the frequencies of occurrence of 115k words were found using the WSJ and the Brown corpora.

In the experiments in this paper we modified the linguistic component to use combinations of different linguistic models. The linguistic component gives the linguistic likelihood of word strings in the sentence. Starting with a baseline LiM (LiM$_B$) we perform LiM adaptation by linearly interpolating the baseline model with other component models trained on different data. The probability of a given n-gram sequence then becomes:

$$P(w_i|w_{i-n+1}..w_{i-1}) = \lambda_1 P_1(w_i|w_{i-n+1}..w_{i-1}) + ... + \lambda_n P_n(w_i|w_{i-n+1}..w_{i-1}), \quad (2)$$

where $\sum_k \lambda_k = 1$ and λ_k and P_k are the weight and the probability assigned by model k.

In the case of a two-sided class-based model,

$$P_k(w_i|w_{i-n+1}..w_{i-1}) = P_k(w_i|C(w_i)) \cdot P_k(C(w_i)|C(w_{i-n+1})..C(w_{i-1})), \quad (3)$$

where $P_k(w_i|C(w_i))$ is the probability of the word w_i belonging to a given class C, and $P_k(C(w_i)|C(w_{i-n+1})..C(w_{i-1}))$ the probability of a certain word class $C(w_i)$ to appear after a history of word classes, $C(w_{i-n+1}), ..., C(w_{i-1})$.

Different types of component LiM are built, coming from different sources of data, either as word or class models. The LiM$_B$ and component LiMs are then combined for adaptation using linear interpolation as in Equation (2). The linguistic score is then computed using this modified probability as in Equation (4):

$$L(w_i) = \log P(w_i|w_{i-n+1}..w_{i-1}). \quad (4)$$

3 Evaluation Criteria

3.1 Summarisation Accuracy

To automatically evaluate the summarised speeches, correctly transcribed talks were manually summarised, and used as the correct targets for evaluation. Variations of manual summarisation results are merged into a word network, which is considered to approximately express all possible correct summarisations covering subjective variations. The word accuracy of automatic summarisation is calculated as the summarisation accuracy (SumACCY) using the word network (Hori et al., 2003):

$$Accuracy = (Len{-}Sub{-}Ins{-}Del)/Len{*}100[\%], \quad (5)$$

where Sub is the number of substitution errors, Ins is the number of insertion errors, Del is the number of deletion errors, and Len is the number of words in the most similar word string in the network.

3.2 ROUGE

Version 1.5.5 of the ROUGE scoring algorithm (Lin, 2004) is also used for evaluating results. ROUGE F-measure scores are given for ROUGE-2 (bigram), ROUGE-3 (trigram), and ROUGE-SU4 (skip-bigram), using the model average (average score across all references) metric.

4 Experimental Setup

Experiments were performed on spontaneous speech, using 9 talks taken from the Translanguage English Database (TED) corpus (Lamel et al., 1994; Wolfel and Burger, 2005), each transcribed and manually summarised by nine different humans for both 10% and 30% summarization ratios. Speech recognition transcriptions (ASR) were obtained for each talk, with an average word error rate of 33.3%.

A corpus consisting of around ten years of conference proceedings (17.8M words) on the subject of speech and signal processing is used to generate the LiM$_B$ and word classes using the clustering algorithm in (Ney et al., 1994).

Different types of component LiM are built and combined for adaptation as described in Section 2.

The first type of component linguistic models are built on the small corpus of hand-made summaries described above, made for the same summarisation ratio as the one we are generating. For each talk the hand-made summaries of the other eight talks (i.e. 72 summaries) were used as the LiM training corpus. This type of LiM is expected to help generate automatic summaries in the same style as those made manually.

		Baseline				Adapted			
		SumACCY	R-2	R-3	R-SU4	SumACCY	R-2	R-3	R-SU4
10%	Random	34.4	0.104	0.055	0.142	-	-	-	-
	Word	63.1	0.186	0.130	0.227	67.8	0.193	0.140	0.228
	Class	65.1	0.195	0.131	0.226	72.6	0.210	0.143	0.234
	Mixed	63.6	0.186	0.128	0.218	71.8	0.211	0.139	0.231
30%	Random	71.2	0.294	0.198	0.331	-	-	-	-
	Word	81.6	0.365	0.271	0.395	83.3	0.365	0.270	0.392
	Class	83.1	0.374	0.279	0.407	92.9	0.415	0.325	0.442
	Mixed	83.1	0.374	0.279	0.407	92.9	0.415	0.325	0.442

Table 1: TRS baseline and adapted results.

The second type of component linguistic models are built from the papers in the conference proceedings for the talk we want to summarise. This type of LiM, used for topic adaptation, is investigated because key words and important sentences that appear in the associated paper are expected to have a high information value and should be selected during the summarisation process.

Three sets of experiments were made: in the first experiment (referred to as Word), LiM_B and both component models are word models, as introduced in (Chatain et al., 2006). For the second one (Class), both LiM_B and the component models are class models built using exactly the same data as the word models. For the third experiment (Mixed), the LiM_B is an interpolation of class and word models, while the component LiMs are class models.

To optimise use of the available data, a rotating form of cross-validation (Duda and Hart, 1973) is used: all talks but one are used for development, the remaining talk being used for testing. Summaries from the development talks are generated automatically by the system using different sets of parameters and the LiM_B. These summaries are evaluated and the set of parameters which maximises the development score for the LiM_B is selected for the remaining talk. The purpose of the development phase is to choose the most effective combination of weights α_C, α_I and α_L. The summary generated for each talk using its set of optimised parameters is then evaluated using the same metric, which gives us our baseline for this talk. Using the same parameters as those that were selected for the baseline, we generate summaries for the lectures in the development set for different LiM interpolation weights λ_k. Values between 0 and 1 in steps of 0.1, were investigated for the latter, and an optimal set of λ_k is selected. Using these interpolation weights, as well as the set of parameters determined for the baseline, we generate a summary of the test talk, which is evaluated using the same evaluation metric, giving us our final adapted result for this talk. Averaging those results over the test set (i.e. all talks) gives us our final adapted result.

This process is repeated for all evaluation metrics, and all three experiments (Word, Class, and Mixed).

Lower bound results are given by random summarisation (Random) i.e. randomly extracting sentences and words, without use of the scores present in Equation (1) for appropriate summarisation ratios.

5 Results

5.1 TRS Results

Initial experiments were made on the human transcriptions (TRS), and results are given in Table 1. Experiments on word models (Word) show relative improvements in terms of SumACCY of 7.5% and 2.1% for the 10% and 30% summarisation ratios, respectively. ROUGE metrics, however, do not show any significant improvement.

Using class models (Class and Mixed), for all ROUGE metrics, relative improvements range from 3.5% to 13.4% for the 10% summarisation ratio, and from 8.6% to 16.5% on the 30% summarisation ratio. For SumACCY, relative improvements between 11.5% to 12.9% are observed.

5.2 ASR Results

ASR results for each experiment are given in Table 2 for appropriate summarisation ratios. As for

		Baseline				Adapted			
		SumACCY	R-2	R-3	R-SU4	SumACCY	R-2	R-3	R-SU4
10%	Random	33.9	0.095	0.042	0.140	-	-	-	-
	Word	48.6	0.143	0.064	0.182	49.8	0.129	0.060	0.173
	Class	50.0	0.133	0.063	0.170	55.1	0.156	0.077	0.193
	Mixed	48.5	0.134	0.068	0.176	56.2	0.142	0.077	0.191
30%	Random	56.1	0.230	0.124	0.283	-	-	-	-
	Word	66.7	0.265	0.157	0.314	68.7	0.271	0.161	0.328
	Class	66.1	0.277	0.165	0.324	71.1	0.300	0.180	0.348
	Mixed	64.9	0.268	0.160	0.312	70.5	0.304	0.192	0.351

Table 2: ASR baseline and adapted results.

the TRS, LiM adaptation showed improvements in terms of SumACCY, but ROUGE metrics do not corroborate those results for the 10% summarisation ratio. Using class models, for all ROUGE metrics, relative improvements range from 6.0% to 22.2% and from 7.4% to 20.0% for the 10% and 30% summarisation ratios, respectively. SumACCY relative improvements range from 7.6% to 15.9%.

6 Discussion

Compared to previous experiments using only word models, improvements obtained using class models are larger and more significant for both ROUGE and SumACCY metrics. This can be explained by the fact that the data we are performing adaptation on is very sparse, and that the nine talks used in these experiments are quite different from each other, especially since the speakers also vary in style. Class models are more robust to this spontaneous speech aspect than word models, since they generalise better to unseen word sequences.

There is little difference between the Class and Mixed results, since the development phase assigned most weight to the class model component in the Mixed experiment, making the results quite similar to those of the Class experiment.

7 Conclusion

In this paper we have investigated linguistic model adaptation using different sources of data for an automatic speech summarisation system. Class models have proved to be much more robust than word models for this process, and relative improvements ranging from 6.0% to 22.2% were obtained on a variety of evaluation metrics on summaries generated from automatic speech recogniser transcriptions.

Acknowledgements: The authors would like to thank M. Wölfel for the recogniser transcriptions and C. Hori for her work on two stage summarisation and gathering the TED corpus data. This work is supported by the 21st Century COE Programme.

References

P. Chatain, E.W.D. Whittaker, J. Mrozinski, and S. Furui. 2006. Topic and Stylistic Adaptation for Speech Summarization. *Proc. ICASSP, Toulouse, France*.

R. Duda and P. Hart. 1973. *Pattern Classification and Scene Analysis*. Wiley, New York.

C. Hori, T. Hori, and S. Furui. 2003. Evaluation Method for Automatic Speech Summarization. *Proc. Eurospeech, Geneva, Switzerland*, 4:2825–2828.

T. Kikuchi, S. Furui, and C. Hori. 2003. Automatic Speech Summarization based on Sentence Extraction and Compaction. *Proc. ICASSP, Hong Kong, China*, 1:236–239.

L. Lamel, F. Schiel, A. Fourcin, J. Mariani, and H. Tillmann. 1994. The Translanguage English Database (TED). *Proc. ICSLP, Yokohama, Japan*, 4:1795–1798.

Chin-Yew Lin. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. *Proc. WAS, Barcelona, Spain*.

H. Ney, U. Essen, and R. Kneser. 1994. On Structuring Probabilistic Dependences in Stochastic Language Modelling. *Computer Speech and Language*, (8):1–38.

M. Wolfel and S. Burger. 2005. The ISL Baseline Lecture Transcription System for the TED Corpus. Technical report, Karlsruhe University.

K. Zechner. 2002. Summarization of Spoken Language-Challenges, Methods, and Prospects. *Speech Technology Expert eZine, Issue.6*.

Semi-supervised Relation Extraction with Label Propagation

Jinxiu Chen[1] **Donghong Ji**[1] **Chew Lim Tan**[2] **Zhengyu Niu**[1]

[1]Institute for Infocomm Research
21 Heng Mui Keng Terrace
119613 Singapore
{jinxiu,dhji,zniu}@i2r.a-star.edu.sg

[2]Department of Computer Science
National University of Singapore
117543 Singapore
tancl@comp.nus.edu.sg

Abstract

To overcome the problem of not having enough manually labeled relation instances for supervised relation extraction methods, in this paper we propose a label propagation (LP) based semi-supervised learning algorithm for relation extraction task to learn from both labeled and unlabeled data. Evaluation on the ACE corpus showed when only a few labeled examples are available, our LP based relation extraction can achieve better performance than SVM and another bootstrapping method.

1 Introduction

Relation extraction is the task of finding relationships between two entities from text. For the task, many machine learning methods have been proposed, including supervised methods (Miller et al., 2000; Zelenko et al., 2002; Culotta and Soresen, 2004; Kambhatla, 2004; Zhou et al., 2005), semi-supervised methods (Brin, 1998; Agichtein and Gravano, 2000; Zhang, 2004), and unsupervised method (Hasegawa et al., 2004).

Supervised relation extraction achieves good performance, but it requires a large amount of manually labeled relation instances. Unsupervised methods do not need the definition of relation types and manually labeled data, but it is difficult to evaluate the clustering result since there is no relation type label for each instance in clusters. Therefore, semi-supervised learning has received attention, which can minimize corpus annotation requirement.

Current works on semi-supervised resolution for relation extraction task mostly use the bootstrapping algorithm, which is based on a **local consistency assumption**: examples close to labeled examples within the same class will have the same labels. Such methods ignore considering the similarity between unlabeled examples and do not perform classification from a global consistency viewpoint, which may fail to exploit appropriate manifold structure in data when training data is limited.

The objective of this paper is to present a label propagation based semi-supervised learning algorithm (LP algorithm) (Zhu and Ghahramani, 2002) for Relation Extraction task. This algorithm works by representing labeled and unlabeled examples as vertices in a connected graph, then propagating the label information from any vertex to nearby vertices through weighted edges iteratively, finally inferring the labels of unlabeled examples after the propagation process converges. Through the label propagation process, our method can make the best of the information of labeled and unlabeled examples to realize a **global consistency assumption**: similar examples should have similar labels. In other words, the labels of unlabeled examples are determined by considering not only the similarity between labeled and unlabeled examples, but also the similarity between unlabeled examples.

2 The Proposed Method

2.1 Problem Definition

Let $X = \{x_i\}_{i=1}^{n}$ be a set of contexts of occurrences of all entity pairs, where x_i represents the contexts of the i-th occurrence, and n is the total number of occurrences of all entity pairs. The first l examples are labeled as y_g ($y_g \in \{r_j\}_{j=1}^{R}$, r_j denotes relation type and R is the total number of relation types). And the remaining $u (u = n - l)$ examples are unlabeled.

Intuitively, if two occurrences of entity pairs have

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 25–28,
New York, June 2006. ©2006 Association for Computational Linguistics

the similar contexts, they tend to hold the same relation type. Based on this assumption, we create a graph where the vertices are all the occurrences of entity pairs, both labeled and unlabeled. The edge between vertices represents their similarity. Then the task of relation extraction can be formulated as a form of propagation on a graph, where a vertex's label propagates to neighboring vertices according to their proximity. Here, the graph is connected with the weights: $W_{ij} = exp(-\frac{s_{ij}^2}{\alpha^2})$, where s_{ij} is the similarity between x_i and x_j calculated by some similarity measures. In this paper,two similarity measures are investigated, i.e. Cosine similarity measure and Jensen-Shannon (JS) divergence (Lin, 1991). And we set α as the average similarity between labeled examples from different classes.

2.2 Label Propagation Algorithm

Given such a graph with labeled and unlabeled vertices, we investigate the label propagation algorithm (Zhu and Ghahramani, 2002) to help us propagate the label information of any vertex in the graph to nearby vertices through weighted edges until a global stable stage is achieved.

Define a $n \times n$ probabilistic transition matrix T $T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{kj}}$, where T_{ij} is the probability to jump from vertex x_j to vertex x_i. Also define a $n \times R$ label matrix Y, where Y_{ij} representing the probabilities of vertex y_i to have the label r_j.

Then the label propagation algorithm consists the following main steps:

Step1: Initialization Firstly, set the iteration index $t = 0$. Then let Y^0 be the initial soft labels attached to each vertex and Y_L^0 be the top l rows of Y^0, which is consistent with the labeling in labeled data ($Y_{ij}^0 = 1$ if y_i is label r_j and 0 otherwise). Let Y_U^0 be the remaining u rows corresponding to unlabeled data points and its initialization can be arbitrary.

Step 2: Propagate the label by $Y^{t+1} = \overline{T}Y^t$, where $\overline{T}$ is the row-normalized matrix of T, i.e. $\overline{T_{ij}} = T_{ij}/\sum_k T_{ik}$, which can maintain the class probability interpretation.

Step 3: Clamp the labeled data, i.e., replace the top l row of Y^{t+1} with Y_L^0. In this step, the labeled data is clamped to replenish the label sources from these labeled data. Thus the labeled data act like sources to push out labels through unlabeled data.

Table 1: Frequency of Relation SubTypes in the ACE training and devtest corpus.

Type	SubType	Training	Devtest
ROLE	General-Staff	550	149
	Management	677	122
	Citizen-Of	127	24
	Founder	11	5
	Owner	146	15
	Affiliate-Partner	111	15
	Member	460	145
	Client	67	13
	Other	15	7
PART	Part-Of	490	103
	Subsidiary	85	19
	Other	2	1
AT	Located	975	192
	Based-In	187	64
	Residence	154	54
SOC	Other-Professional	195	25
	Other-Personal	60	10
	Parent	68	24
	Spouse	21	4
	Associate	49	7
	Other-Relative	23	10
	Sibling	7	4
	GrandParent	6	1
NEAR	Relative-Location	88	32

Step 4: Repeat from step 2 until Y converges.

Step 5: Assign $x_h (l + 1 \leq h \leq n)$ with a label: $y_h = argmax_j Y_{hj}$.

3 Experiments and Results

3.1 Data

Our proposed graph-based method is evaluated on the ACE corpus [1], which contains 519 files from sources including broadcast, newswire, and newspaper. A break-down of the tagged data by different relation subtypes is given in Table 1.

3.2 Features

We extract the following lexical and syntactic features from two entity mentions, and the contexts before, between and after the entity pairs. Especially, we set the mid-context window as everything between the two entities and the pre- and post- context as up to two words before and after the corresponding entity. Most of these features are computed from the parse trees derived from Charniak Parser (Charniak, 1999) and the Chunklink script [2] written by Sabine Buchholz from Tilburg University.

[1] http://www.ldc.upenn.edu/Projects/ACE/

[2] Software available at http://ilk.uvt.nl/∼sabine/chunklink/

Table 2: Performance of Relation Detection: SVM and LP algorithm with different size of labeled data. The LP algorithm is performed with two similarity measures: Cosine similarity and JS divergence.

	SVM			LP_{Cosine}			LP_{JS}		
Percentage	P	R	F	P	R	F	P	R	F
1%	35.9	32.6	34.4	58.3	56.1	57.1	58.5	58.7	58.5
10%	51.3	41.5	45.9	64.5	57.5	60.7	64.6	62.0	63.2
25%	67.1	52.9	59.1	68.7	59.0	63.4	68.9	63.7	66.1
50%	74.0	57.8	64.9	69.9	61.8	65.6	70.1	64.1	66.9
75%	77.6	59.4	67.2	71.8	63.4	67.3	72.4	64.8	68.3
100%	79.8	62.9	70.3	73.9	66.9	70.2	74.2	68.2	71.1

Table 3: Performance of Relation Classification on Relation Subtype: SVM and LP algorithm with different size of labeled data. The LP algorithm is performed with two similarity measures: Cosine similarity and JS divergence.

	SVM			LP_{Cosine}			LP_{JS}		
Percentage	P	R	F	P	R	F	P	R	F
1%	31.6	26.1	28.6	39.6	37.5	38.5	40.1	38.0	39.0
10%	39.1	32.7	35.6	45.9	39.6	42.5	46.2	41.6	43.7
25%	49.8	35.0	41.1	51.0	44.5	47.3	52.3	46.0	48.9
50%	52.5	41.3	46.2	54.1	48.6	51.2	54.9	50.8	52.7
75%	58.7	46.7	52.0	56.0	52.0	53.9	56.1	52.6	54.3
100%	60.8	48.9	54.2	56.2	52.3	54.1	56.3	52.9	54.6

Words: Surface tokens of the two entities and three context windows.

Entity Type: the entity type of both entity mentions, which can be PERSON, ORGANIZATION, FACILITY, LOCATION and GPE.

POS: Part-Of-Speech tags corresponding to all tokens in the two entities and three context windows.

Chunking features: Chunk tag information and Grammatical function of the two entities and three context windows. IOB-chains of the heads of the two entities are also considered. IOB-chain notes the syntactic categories of all the constituents on the path from the root node to this leaf node of tree.

We combine the above features with their position information in the context to form the context vector. Before that, we filter out low frequency features which appeared only once in the entire set.

Table 4: Comparison of performance on individual relation type of Zhang (2004)'s method and our method. For Zhang (2004)'s method, feature sampling probability is set to 0.3 and agreement threshold is set to 9 out of 10.

	Bootstrapping			LP_{JS}		
Rel-Type	P	R	F	P	R	F
ROLE	78.5	69.7	73.8	81.0	74.7	77.7
PART	65.6	34.1	44.9	70.1	41.6	52.2
AT	61.0	84.8	70.9	74.2	79.1	76.6
SOC	47.0	57.4	51.7	45.0	59.1	51.0
NEAR	$undef$	0	$undef$	13.7	12.5	13.0

3.3 Experimental Evaluation

3.3.1 Relation Detection

We collect all entity mention pairs which co-occur in the same sentence from the training and devtest corpus into two set $C1$ and $C2$ respectively. The set $C1$ includes annotated training data $AC1$ and unrelated data $UC1$. We randomly sample l examples from $AC1$ as **labeled data** and add a “NONE” class into labeled data for the case where the two entity mentions are not related. The data of the “NONE” class is resulted by sampling l examples from $UC1$. Moreover, we combine the rest examples of $C1$ and the whole set $C2$ as **unlabeled data**.

Given labeled and unlabeled data,we can perform LP algorithm to detect possible relations, which are those entity pairs that are not classified to the “NONE” class but to the other 24 subtype classes. In addition,we conduct experiments with different sampling set size l, including $1\% \times N_{train}$,$10\% \times N_{train}$,$25\% \times N_{train}$,$50\% \times N_{train}$,$75\% \times N_{train}$, $100\% \times N_{train}$ ($N_{train} = |AC1|$). If any major subtype was absent from the sampled labeled set,we redo the sampling. For each size,we perform 20 trials and calculate an average of 20 random trials.

3.3.2 SVM vs. LP

Table 2 reports the performance of relation detection by using SVM and LP with different sizes of labled data. For SVM, we use LIBSVM tool with linear kernel function [3]. And the same sampled labeled data used in LP is used to train SVM models. From Table 2, we see that both LP$_{Cosine}$ and LP$_{JS}$ achieve higher *Recall* than SVM. Especially, with small labeled dataset (percentage of labeled data $\leq$ 25%), this merit is more distinct. When the percentage of labeled data increases from 50% to 100%, LP$_{Cosine}$ is still comparable to SVM in *F-measure* while LP$_{JS}$ achieves better *F-measure* than SVM. On the other hand, LP$_{JS}$ consistently outperforms LP$_{Cosine}$.

Table 3 reports the performance of relation classification, where the performance describes the average values over major relation subtypes. From Table 3, we see that LP$_{Cosine}$ and LP$_{JS}$ outperform SVM by *F-measure* in almost all settings of labeled data, which is due to the increase of *Recall*. With smaller labeled dataset, the gap between LP and SVM is larger. On the other hand, LP$_{JS}$ divergence consistently outperforms LP$_{Cosine}$.

3.3.3 LP vs. Bootstrapping

In (Zhang, 2004), they perform relation classification on ACE corpus with bootstrapping on top of SVM. To compare with their proposed Bootstrapped SVM algorithm, we use the same feature stream setting and randomly selected 100 instances from the training data as the size of initial labeled data.

Table 4 lists the performance on individual relation type. We can find that LP algorithm achieves 6.8% performance improvement compared with the (Zhang, 2004)'s bootstrapped SVM algorithm average on all five relation types. Notice that performance reported on relation type "NEAR" is low, because it occurs rarely in both training and test data.

4 Conclusion and Future work

This paper approaches the task of semi-supervised relation extraction on Label Propagation algorithm. Our results demonstrate that, when only very few labeled examples are available, this manifold learning based algorithm can achieve better performance than supervised learning method (SVM) and bootstrapping based method, which can contribute to minimize corpus annotation requirement. In the future we would like to investigate how to select more useful feature stream and whether feature selection method can improve the performance of our graph-based semi-supervised relation extraction.

[3] $LIBSVM$: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

References

Agichtein E. and Gravano L. 2000. *Snowball: Extracting Relations from large Plain-Text Collections, In Proceeding of the* 5^{th} *ACM International Conference on Digital Libraries*.

Brin Sergey. 1998. *Extracting patterns and relations from world wide web. In Proceeding of WebDB Workshop at 6th International Conference on Extending Database Technology*. pages 172-183.

Charniak E. 1999. *A Maximum-entropy-inspired parser. Technical Report CS-99-12*. Computer Science Department, Brown University.

Culotta A. and Soresen J. 2004. *Dependency tree kernels for relation extraction, In Proceedings of 42th ACL conference*.

Hasegawa T., Sekine S. and Grishman R. 2004. *Discovering Relations among Named Entities from Large Corpora, In Proceeding of Conference ACL2004*. Barcelona, Spain.

Kambhatla N. 2004. *Combining lexical, syntactic and semantic features with Maximum Entropy Models for extracting relations, In Proceedings of 42th ACL conference*. Spain.

Lin,J. 1991. *Divergence Measures Based on the Shannon Entropy. IEEE Transactions on Information Theory*. 37:1,145-150.

Miller S.,Fox H.,Ramshaw L. and Weischedel R. 2000. *A novel use of statistical parsing to extract information from text. In Proceedings of 6th Applied Natural Language Processing Conference* 29 April-4 may 2000, Seattle USA.

Yarowsky D. 1995. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. pp.189-196.

Zelenko D., Aone C. and Richardella A. 2002. *Kernel Methods for Relation Extraction, In Proceedings of the EMNLP Conference*. Philadelphia.

Zhang Zhu. 2004. *Weakly-supervised relation classification for Information Extraction, In proceedings of ACM 13th conference on Information and Knowledge Management*. 8-13 Nov 2004. Washington D.C.,USA.

Zhou GuoDong, Su Jian, Zhang Jie and Zhang min. 2005. *Combining lexical, syntactic and semantic features with Maximum Entropy Models for extracting relations, In proceedings of 43th ACL conference*. USA.

Zhu Xiaojin and Ghahramani Zoubin. 2002. *Learning from Labeled and Unlabeled Data with Label Propagation. CMU CALD tech report CMU-CALD-02-107*.

Temporal Classification of Text and Automatic Document Dating

Angelo Dalli
University of Sheffield
211, Portobello Street
Sheffield, S1 4DP, UK
angelo@dcs.shef.ac.uk

Abstract

Temporal information is presently under-utilised for document and text processing purposes. This work presents an unsupervised method of extracting periodicity information from text, enabling time series creation and filtering to be used in the creation of sophisticated language models that can discern between repetitive trends and non-repetitive writing pat-terns. The algorithm performs in O(n log n) time for input of length n. The temporal language model is used to create rules based on temporal-word associations inferred from the time series. The rules are used to automatically guess at likely document creation dates, based on the assumption that natural languages have unique signatures of changing word distributions over time. Experimental results on news items spanning a nine year period show that the proposed method and algorithms are accurate in discovering periodicity patterns and in dating documents automatically solely from their content.

1 Introduction

Various features have been used to classify and predict the characteristics of text and related text documents, ranging from simple word count models to sophisticated clustering and Bayesian models that can handle both linear and non-linear classes. The general goal of most classification research is to assign objects from a pre-defined domain (such as words or entire documents) to two or more classes/categories. Current and past research has largely focused on solving problems like tagging, sense disambiguation, sentiment classification, author and language identification and topic classification. We introduce an unsupervised method that classifies text and documents according to their predicted time of writing/creation. The method uses a sophisticated temporal language model to predict likely creation dates for a document, hence dating it automatically. This short paper presents some background information about existing techniques and the implemented system, followed by a brief explanation of the classification and dating method, and finally concluding with results and evaluation performed on the LDC GigaWord English Corpus (LDC, 2003).

2 Background

Temporal information is presently under-utilised for document and text processing purposes. Past and ongoing research work has largely focused on the identification and tagging of temporal expressions, with the creation of tagging methodologies such as TimeML/TIMEX (Gaizauskas and Setzer, 2002; Pustejovsky et al., 2003; Ferro et al., 2004), TDRL (Aramburu and Berlanga, 1998) and associated evaluations such as the ACE TERN competition (Sundheim et al. 2004).

Temporal analysis has also been applied in Question-Answering systems (Pustejovsky et al., 2004; Schilder and Habel, 2003; Prager et al., 2003), email classification (Kiritchenko et al.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 29–32,
New York, June 2006. ©2006 Association for Computational Linguistics

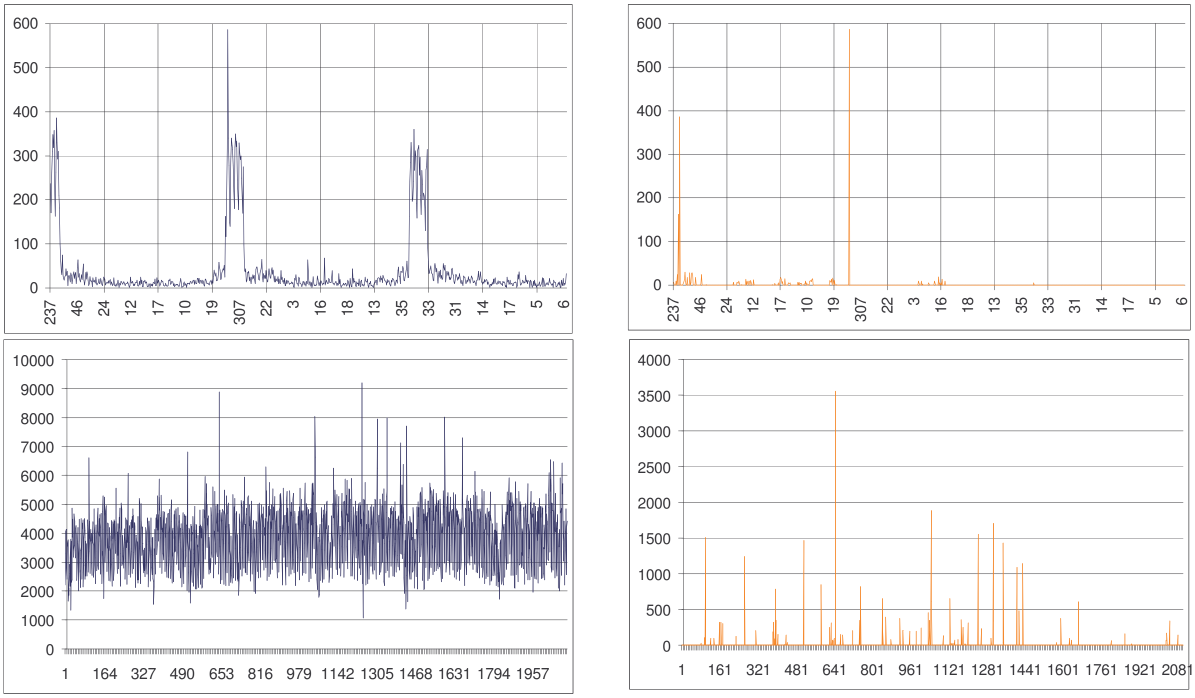


Figure 1 Effects of applying the temporal periodical algorithm on time series for "January" (top) and "the" (bottom) with original series on the left and the remaining time series component after filtering on the right. Y-axis shows frequency count and X-axis shows the day number (time).

2004), aiding the precision of Information Retrieval results (Berlanga et al., 2001), document summarisation (Mani and Wilson, 2000), time stamping of event clauses (Filatova and Hovy, 2001), temporal ordering of events (Mani et al., 2003) and temporal reasoning from text (Boguraev and Ando, 2005; Moldovan et al., 2005). There is also a large body of work on time series analysis and temporal logic in Physics, Economics and Mathematics, providing important techniques and general background information. In particular, this work uses techniques adapted from Seasonal Auto-Regressive Integrated Moving Average models (SARIMA). SARIMA models are a class of seasonal, non-stationary temporal models based on the ARIMA process (defined as a non-stationary extension of the stationary ARMA model). Non-stationary ARIMA processes are defined by:

$$(1-B)^d \phi(B) X_t = \theta(B) Z_t \qquad (1)$$

where d is non-negative integer, and $\phi(X)$ $\theta(X)$ polynomials of degrees p and q respectively. The exact parameters for each process (one process per word) are determined automatically by the system. A discussion of the general SARIMA model is beyond the scope of this paper (details can be found in Mathematics & Physics publications). The NLP application of temporal classification and prediction to guess at likely document and text creation dates is a novel application that has not been considered much before, if at all.

3 Temporal Periodicity Analysis

We have created a high-performance system that decomposes time series into two parts: a periodic component that repeats itself in a predictable manner, and a non-periodic component that is left after the periodic component has been filtered out from the original time series. Figure 1 shows an example of the filtering results on time-series of the words “January” and “the”. The time series are based on training documents selected at random from the GigaWord English corpus. 10% of all the documents in the corpus were used as training documents, with the rest being available for evaluation and testing. A total of 395,944 time series spanning 9 years were calculated from the GigaWord corpus. Figure 2 presents pseudo-code for the time series decomposition algorithm:

```
1. Find min/max/mean and standard deviation of time series
2. Start with a pre-defined maximum window size (presently set to 366 days)
3. While window size bigger than 1 repeat steps a. to d. below:
    a. Look at current value in time series (starting first value)
    b. Do values at positions current, current + window size, current + 2 x window size, etc. vary by less than ½ standard deviation?
    c. If yes, mark current value/window size pair as being possible decomposition match
    d. Look at next value in time series until the end is reached
    e. Decrease window size by one
4. Select the minimum number of decomposition matches that cover the entire time series using a greedy algorithm
```

Figure 2 Time Series Decomposition Algorithm

The time series decomposition algorithm was applied to the 395,944 time series, taking an average of 419ms per series. The algorithm runs in $O(n \log n)$ time for a time series of length n.

The periodic component of the time series is then analysed to extract temporal association rules between words and different "seasons", including Day of Week, Week Number, Month Number, Quarter, and Year. The procedure of determining if a word, for example, is predominantly peaking on a weekly basis, is to apply a sliding window of size 7 (in the case of weekly periods) and determining if the periodic time series always spikes within this window. Figure 3 shows the frequency distribution of the periodic time series component of the days of week names ("Monday", "Tuesday", etc.) Note that the frequency counts peak exactly on that particular day of the week. For example, the word "Monday" is automatically associated with Day 1, and "April" associated with Month 4. The creation of temporal association rules generalises inferences obtained from the periodic data. Each association rule has the following information:

Word ID
Period Type (Week, Month, etc.)
Period Number and Score Matrix

The period number and score matrix represent a probability density function that shows the likelihood of a word appearing on a particular period number. For example, the score matrix for "January" will have a high score for period 1 (and period type set to Monthly). Figure 4 shows some examples of extracted association rules. The PDF scores are shown in Figure 4 as they are stored internally (as multiples of the standard deviation of that time series) and are automatically normalised during the classification process at runtime. Rule generalisation is not possible in such a straightforward manner for the non-periodic data. The use of non-periodic data to optimise the results of the temporal classification and automatic dating system is not covered in this paper.

4 Temporal Classification and Dating

The periodic temporal association rules are utilised to automatically guess at the creation date of documents automatically. Documents are input into the system and the probability density functions for each word are weighted and added up. Each PDF is weighted according to the inverse document frequency (IDF) of each associated word. Periods that obtain high score are then ranked for each type of period and two guesses per period type are obtained for each document. Ten guesses in total are thus obtained for Day of Week, Week Number, Month Number, Quarter, and Year (5 period types x 2 guesses each).

	Su	M	T	W	Th	F	S
0	**22660**	10540	7557	772	2130	3264	11672
1	12461	**37522**	10335	6599	1649	3222	3414
2	3394	18289	**38320**	9352	7300	2543	2261
3	2668	4119	18120	**36933**	10427	5762	2147
4	2052	2602	3910	17492	**36094**	9098	5667
5	5742	1889	2481	2568	17002	**32597**	7849
6	7994	7072	1924	1428	3050	14087	**21468**
Av	8138	11719	11806	10734	11093	10081	7782
St	7357	12711	12974	12933	12308	10746	6930

Figure 3 Days of Week Temporal Frequency Distribution for extracted Periodic Component displayed in a Weekly Period Type format

```
January
Week    1       2       3       4       5
Score   1.48    2.20    3.60    3.43    3.52
Month   1       Score 2.95
Quarter         1       Score 1.50

Christmas
Week    2       5       36      42      44
Score   1.32    0.73    1.60    0.83    1.32
```

```
Week   47      49      50      51      52
Score  1.32    2.20    2.52    2.13    1.16

Month  1       9       10      11      12
Score  1.10    0.75    1.63    1.73    1.98
Quarter        4       Score 1.07
```

Figure 4 Temporal Classification Rules for Periodic Components of "January" and "Christmas"

5 Evaluation, Results and Conclusion

The system was trained using 67,000 news items selected randomly from the GigaWord corpus. The evaluation took place on 678,924 news items extracted from items marked as being of type “story” or “multi”. Table 1 presents a summary of results. Processing took around 2.33ms per item.

Type	Correct	Incorrect	Avg. Error
DOW	218,899 (32.24%)	460,025 (67.75%)	1.89 days
Week	24,660 (3.53%)	654,264 (96.36%)	14.37 wks
Month	122,777 (18.08%)	556,147 (81.91%)	2.57 mths
Quarter	337,384 (49.69%)	341,540 (50.30%)	1.48 qts
Year	596,009 (87.78%)	82,915 (12.21%)	1.74 yrs
Combined	**422,358 (62.21%)**	**256,566 (37.79%)**	**210 days**

Table 1 Evaluation Results Summary

The actual date was extracted from each news item in the GigaWord corpus and the day of week (DOW), week number and quarter calculated from the actual date. Average errors for each type of classifier were calculated automatically. For results to be considered correct, the system had to have the predicted value ranked in the first position equal to the actual value (of the type of period). The system results show that reasonable accurate dates can be guessed at the quarterly and yearly levels. The weekly classifier had the worst performance of all classifiers. The combined classifier uses a simple weighted formula to guess the final document date using input from all classifiers. The weights for the combined classifier have been set on the basis of this evaluation. The temporal classification and analysis system presented in this paper can handle any Indo-European language in its present form. Further work is being carried out to extend the system to Chinese and Arabic. Current research is aiming at improving the accuracy of the classifier by using the non-periodic components and improving the combined classification method.

References

Aramburu, M. Berlanga, R. 1998. *A Retrieval Language for Historical Documents*. LNCS, 1460, pp. 216-225.

Berlanga, R. Perez, J. Aramburu, M. Llido, D. 2001. *Techniques and Tools for the Temporal Analysis of Retrieved Information*. LNCS, 2113, pp. 72-81.

Boguraev, B. Ando, R.K. 2005. *TimeML-Compliant Text Analysis for Temporal Reasoning*. IJCAI-2005.

Ferro, L. Gerber, L. Mani, I. Sundheim, B. Wilson, G. 2004. *TIDES Standard for the Annotation of Temporal Expressions*. The MITRE Corporation.

Filatova, E. Hovy, E. 2001. *Assigning time-stamps to event-clauses*. Proc. EACL 2001, Toulouse, France.

Gaizauskas, R. Setzer, A. 2002. *Annotation Standards for Temporal Information in NL*. Proc. LREC 2002.

Kiritchenko, S. Matwin, S. Abu-Hakima, S. 2004. *Email Classification with Temporal Features*. Proc. IIPWM 2004, Zakopane, Poland. pp. 523-534.

Linguistic Data Consortium (LDC). 2003. English Gigaword Corpus. David Graff, ed. LDC2003T05.

Mani, I. Wilson, G. 2000. *Robust temporal processing of news*. Proc. ACL 2000, Hong Kong.

Mani, I. Schiffman, B. Zhang, J. 2003. *Inferring temporal ordering of events in news*. HLT-NAACL 2003.

Moldovan, D. Clark, C. Harabagiu, S. 2005. *Temporal Context Representation and Reasoning*. IJCAI-2005.

Prager, J. Chu-Carroll, J. Brown, E. Czuba, C. 2003. *Question Answering using predictive annotation*.

Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, R. Setzer, A. Katz, G. 2003. *TimeML: Robust Specification of event and temporal expressions in text*. IWCS-5.

Pustejovsky, J. Sauri, R. Castano, J. Radev, D. Gaizauskas, R. Setzer, A. Sundheim, B. Katz, G. 2004. “Representing Temporal and Event Knowledge for QA Systems”. *New Directions in QA*, MIT Press.

Schilder, F. Habel, C. 2003. *Temporal Information Extraction for Temporal QA*. AAAI NDQA, pp. 35-44.

Sundheim, B. Gerber, L. Ferro, L. Mani, I. Wilson, G. 2004. *Time Expression Recognition and Normalization (TERN)*. http://timex2.mitre.org.

Answering the Question You Wish They Had Asked:
The Impact of Paraphrasing for Question Answering

Pablo Ariel Duboue
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532, USA
duboue@us.ibm.com

Jennifer Chu-Carroll
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532, USA
jencc@us.ibm.com

Abstract

State-of-the-art Question Answering (QA) systems are very sensitive to variations in the phrasing of an information need. Finding the preferred language for such a need is a valuable task. We investigate that claim by adopting a simple MT-based paraphrasing technique and evaluating QA system performance on paraphrased questions. We found a potential increase of 35% in MRR with respect to the original question.

1 Introduction

In a typical Question Answering system, an input question is analyzed to formulate a query to retrieve relevant documents from a target corpus (Chu-Carroll et al., 2006; Harabagiu et al., 2006; Sun et al., 2006). This analysis of the input question affects the subset of documents that will be examined and ultimately plays a key role in determining the answers the system chooses to produce. However, most existing QA systems, whether they adopt knowledge-based, statistical, or hybrid methods, are very sensitive to small variations in the question form, often yielding substantially different answers for questions that are semantically equivalent. For example, our system's answer to *"Who invented the telephone?"* is *"Alexander Graham Bell;"* however, its top answer to a paraphrase of the above question *"Who is credited with the invention of the telephone?"* is *"Gutenberg,"* who is credited with the invention of the printing press, while *"Alexander Graham Bell,"* who is credited with the invention of the telephone, appears in rank four.

To demonstrate the ubiquity of this phenomenon, we asked the aforementioned two questions to several QA systems on the web, including LCC's PowerAnswer system,[1] MIT's START system,[2] AnswerBus,[3] and Ask Jeeves.[4] All systems exhibited different behavior for the two phrasings of the question, ranging from minor variations in documents presented to justify an answer, to major differences such as the presence of correct answers in the answer list. For some systems, the more complex question form posed sufficient difficulty that they chose not to answer it.

In this paper we focus on investigating a high risk but potentially high payoff approach, that of improving system performance by **replacing** the user question with a paraphrased version of it. To obtain candidate paraphrases, we adopt a simple yet powerful technique based on machine translation, which we describe in the next section. Our experimental results show that we can potentially achieve a 35% relative improvement in system performance if we have an oracle that always picks the optimal paraphrase for each question. Our ultimate goal is to automatically select from the set of candidates a high potential paraphrase using a component trained against the QA system. In Section 3, we present our initial approach to paraphrase selection which shows that, despite the tremendous odds against selecting performance-improving paraphrases, our conservative selection algorithm resulted in marginal improvement in system performance.

[1] http://www.languagecomputer.com/demos
[2] http://start.csail.mit.edu
[3] http://www.answerbus.com
[4] http://www.ask.com

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 33–36,
New York, June 2006. ©2006 Association for Computational Linguistics

(A)	*What toxins are most* **hazardous** *to* **expectant mothers***?*	en→it	*Che tossine sono più pericolose alle donne incinte?*	it→en	*Which toxins are more* **dangerous** *to the* **pregnant women***?*
(B)	*Find out about* **India's nuclear weapons program**.	en→es	*Descubra sobre el programa de las armas nucleares de la India.*	es→en	*Discover on the* **program of the nuclear weapons of India**.

Figure 1: Example of lexical and syntactical paraphrases via MT-paraphrasing using Babelfish.

2 MT-Based Automatic Paraphrasing

To measure the impact of paraphrases on QA systems, we seek to adopt a methodology by which paraphrases can be automatically generated from a user question. Inspired by the use of parallel translations to mine paraphrasing lexicons (Barzilay and McKeown, 2001) and the use of MT engines for word sense disambiguation (Diab, 2000), we leverage existing machine translation systems to generate semantically equivalent, albeit lexically and syntactically distinct, questions.

Figure 1 (A) illustrates how MT-based paraphrasing captures lexical paraphrasing, ranging from obtaining simple synonyms such as *hazardous* and *dangerous* to deriving more complex equivalent phrases such as *expectant mother* and *pregnant woman*. In addition to lexical paraphrasing, some two-way translations achieve structural paraphrasing, as illustrated by the example in Figure 1 (B).

Using multiple MT engines can help paraphrase diversity. For example, in Figure 1 (B), if we use the @promt translator[5] for English-to-Spanish translation and Babelfish[6] for Spanish-to-English translation, we get *"Find out on the* **nuclear armament program of India***"* where both lexical and structural paraphrasings are observed.

The motivation of generating an array of lexically and structurally distinct paraphrases is that some of these paraphrases may better match the processing capabilities of the underlying QA system than the original question and are thus more likely to produce correct answers. Our observation is that while the paraphrase set contains valuable performance-improving phrasings, it also includes a large number of ungrammatical sentences which need to be filtered out to reduce negative impact on performance.

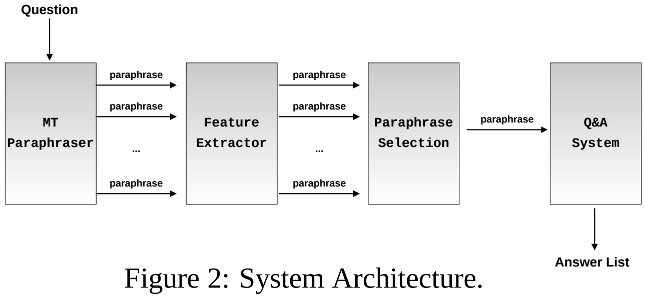


Figure 2: System Architecture.

3 Using Automatic Paraphrasing in Question Answering

We use a generic architecture (Figure 2) that treats a QA system as a black box that is invoked after a paraphrase generation module, a feature extraction module, and a paraphrase selection module are executed. The preprocessing modules identifies a paraphrase of the original question, which could be the question itself, to send as input to the QA system. A key advantage of treating the core QA system as a black box is that the preprocessing modules can be easily applied to improve the performance of any QA system.[7]

We described the paraphrase generation module in the previous section and will discuss the remaining two modules below.

Feature Extraction Module. For each possible paraphrase, we compare it against the original question and compute the features shown in Table 1. These are a subset of the features that we have experimented with and have found to be meaningful for the task. All of these features are required in or-

[5] http://www.online-translator.com

[6] http://babelfish.altavista.com

[7] In our earlier experiments, we adopted an approach that combines answers to all paraphrases through voting. These experiments proved unsuccessful: in most cases, the answer to the original question was amplified, both when right and wrong.

Feature	Description	Intuition
Sum IDF	The sum of the IDF scores for all terms in the original question and the paraphrase.	Paraphrases with more informative terms for the corpus at hand should be preferred.
Lengths	Number of query terms for each of the paraphrase and the original question.	We expect QA systems to prefer shorter paraphrases.
Cosine Distance	The distance between the vectors of both questions, IDF-weighted.	Certain paraphrases diverge too much from the original.
Answer Types	Whether answer types, as predicted by our question analyzer, are the same or overlap.	Choosing a paraphrase that does not share an answer type with the original question is risky.

Table 1: Our features, computed for each paraphrase by comparing it against the original question.

der not to lower the performance with respect to the original question. They are ordered by their relative contributions to the error rate reduction.

Paraphrase Selection Module. To select a paraphrase, we used JRip, the Java re-implementation of `ripper` (Cohen, 1996), a supervised rule learner in the Weka toolkit (Witten and Frank, 2000).

We initially formulated paraphrase selection as a three-way classification problem, with an attempt to label each paraphrase as being "worse," the "same," or "better" than the original question. Our objective was to **replace** the original question with a paraphrase labeled "better." However, the priors for these classes are roughly 30% for "worse," 65% for "same," and 5% for "better". Our empirical evidence shows that successfully pinpointing a "better" paraphrase improves, on average, the reciprocal rank for a question by 0.5, while erroneously picking a "worse" paraphrase results in a 0.75 decrease. That is to say, errors are 1.5 times more costly than successes (and five times more likely). This scenario strongly suggests that a high precision algorithm is critical for this component to be effective.

To increase precision, we took two steps. First, we trained a cascade of two binary classifiers. The first one classifies "worse" versus "same or better," with a bias for "worse." The second classifier has classes "worse or same" versus "better," now with a bias towards "better." The second step is to constrain the confidence of the classifier and only accept paraphrases where the second classifier has a 100% confidence. These steps are necessary to avoid decreasing performance with respect to the original question, as we will show in the next section.

4 Experimental Results

We trained the paraphrase selection module using our QA system, PIQUANT (Chu-Carroll et al., 2006). Our target corpus is the AQUAINT corpus, employed in the TREC QA track since 2002.

As for MT engines, we employed Babelfish and Google MT,[8] rule-based systems developed by SYSTRAN and Google, respectively. We adopted different MT engines based on the hypothesis that differences in their translation rules will improve the effectiveness of the paraphrasing module.

To measure performance, we trained and tested by cross-validation over 712 questions from the TREC 9 and 10 datasets. We paraphrased the questions using the four possible combinations of MT engines with up to 11 intermediate languages, obtaining a total of 15,802 paraphrases. These questions were then fed to our system and evaluated per TREC answer key. We obtained a baseline MRR (top five answers) of 0.345 running over the original questions. An oracle run, in which the best paraphrase (or the original question) is always picked would yield a MRR of 0.48. This potential increase is substantial, taking into account that a 35% improvement separated the tenth participant from the second in TREC-9. Our three-fold cross validation using the features and algorithm described in Section 3 yielded a MRR of 0.347. Over 712 questions, it replaced 14, two of which improved performance, the rest stayed the same. On the other hand, random selection of paraphrases decreased performance to 0.156, clearly showing the importance of selecting a good paraphrase.

[8] `http://translate.google.com`

5 Related Work

Most of the work in QA and paraphrasing focused on folding paraphrasing knowledge into the question analyzer or the answer locator (Rinaldi et al., 2003; Tomuro, 2003). Our work, on the contrary, focuses on question paraphrasing as an external component, independent of the QA system architecture.

Some authors (Dumais et al., 2002; Echihabi et al., 2004) considered the query sent to a search engine as a "paraphrase" of the original natural language question. For instance, Echihabi et al. (2004) presented a large number of "reformulations" that transformed the query into assertions that could match the answers in text. Here we understand a question paraphrase as a reformulation that is itself a question, not a search engine query.

Other efforts in using paraphrasing for QA (Duclaye et al., 2003) focused on using the Web to obtain different verbalizations for a seed relation (e.g., Author/Book); however, they have yet to apply their learned paraphrases to QA.

Recently, there has been work on identifying paraphrases equivalence classes for log analysis (Hedstrom, 2005). Hedstrom used a vector model from Information Retrieval that inspired our cosine measure feature described in Section 3.

6 Conclusions

The work presented here makes contributions at three different levels. First, we have shown that potential impact of paraphrasing with respect to QA performance is significant. Replacing a question with a more felicitously worded question can potentially result in a 35% performance increase.

Second, we performed our experiments by tapping into a readily available paraphrase resource: MT engines. Our results speak of the usefulness of the approach in producing paraphrases. This technique of obtaining a large, although low quality, set of paraphrases can be easily employed by other NLP practitioners wishing to investigate the impact of paraphrasing on their own problems.

Third, we have shown that the task of selecting a better phrasing is amenable to learning, though more work is required to achieve its full potential. In that respect, the features and architecture discussed in Section 3 are a necessary first step in that direction.

In future work, we are interested in developing effective filtering techniques to reduce our candidate set to a small number of high precision paraphrases, in experimenting with state-of-the-art paraphrasers, and in using paraphrasing to improve the stability of the QA system.

Acknowledgments

The authors would like to thank Nelson Correa and Annie Ying for helpful discussions and comments. This work was supported in part by the Disruptive Technology Office (DTO)'s Advanced Question Answering for Intelligence (AQUAINT) Program under contract number H98230-04-C-1577.

References

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-EACL 2001)*, Toulouse, France, July.

Jennifer Chu-Carroll, Pablo A. Duboue, John M. Prager, and Krzysztof Czuba. 2006. IBM's piquant II in TREC 2005. In E. M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Fourthteen Text REtrieval Conference Proceedings (TREC 2005)*, Gaithersburg, MD, USA.

William Cohen. 1996. Learning trees and rules with set-valued features. In *Proceedings of the 14th joint American Association for Artificial Intelligence and IAAI Conference (AAAI/IAAI-96)*, pages 709–716. American Association for Artificial Intelligence.

Mona Diab. 2000. An unsupervised method for word sense tagging using parallel corpora: A preliminary investigation. In *Special Interest Group in Lexical Semantics (SIGLEX) Workshop, Association for Computational Linguistics*, Hong Kong, China, October.

Florence Duclaye, Francois Yvon, and Olivier Collin. 2003. Learning paraphrases to improve a question-answering system. In *EACL 2003, 11th Conference of the European Chapter of the Association for Computational Linguistics, Workshop in NLP for QA*, Budapest, Hungary, April.

S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. 2002. Web question answering: is more always better? In *Proc. SIGIR '02*, pages 291–298, New York, NY, USA. ACM Press.

A. Echihabi, U.Hermjakob, E. Hovy, D. Marcu, E. Melz, and D. Ravichandran. 2004. Multiple-engine question answering in textmap. In *Proc. TREC 2003*.

S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, A. Hickl, and P. Wang. 2006. Employing two question answering systems. In *Proc. TREC 2005*.

Anna Hedstrom. 2005. Question categorization for a question answering system using a vector space model. Master's thesis, Department of Linguistics and Philology (Language Technology Programme) Uppsala University, Uppsala, Sweden.

Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. 2003. Exploiting paraphrases in a question answering system. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 25–32, July.

R. Sun, J. Jiang, Y.F. Tan, H. Cui, T.-S. Chua, and M.-Y. Kan. 2006. Using syntactic and semantic relation analysis in question answering. In *Proc. TREC 2005*.

Noriko Tomuro. 2003. Interrogative reformulation patterns and acquisition of question paraphrases. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 33–40, July.

Ian H. Witten and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers.

Gesture Improves Coreference Resolution

Jacob Eisenstein and Randall Davis
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
{jacobe+davis}@csail.mit.edu

Abstract

Coreference resolution, like many problems in natural language processing, has most often been explored using datasets of written text. While spontaneous spoken language poses well-known challenges, it also offers additional modalities that may help disambiguate some of the inherent disfluency. We explore features of hand gesture that are correlated with coreference. Combining these features with a traditional textual model yields a statistically significant improvement in overall performance.

1 Introduction

Although the natural language processing community has traditionally focused largely on text, face-to-face spoken language is ubiquitous, and offers the potential for breakthrough applications in domains such as meetings, lectures, and presentations. We believe that in face-to-face discourse, it is important to consider the possibility that non-verbal communication may offer features that are critical to language understanding. However, due to the long-standing emphasis on text datasets, there has been relatively little work on non-textual features in unconstrained natural language (prosody being the most notable exception).

Multimodal research in NLP has typically focused on dialogue systems for human-computer interaction (e.g., (Oviatt, 1999)); in contrast, we are interested in the applicability of multimodal features to unconstrained human-human dialogues. We believe that such features will play an essential role in bringing NLP applications such as automatic summarization and segmentation to multimedia documents, such as lectures and meetings.

More specifically, in this paper we explore the possibility of applying hand gesture features to the problem of coreference resolution, which is thought to be fundamental to these more ambitious applications (Baldwin and Morton, 1998). To motivate the need for multimodal features in coreference resolution, consider the following transcript:

> "[This circle (1)] is rotating clockwise and [this piece of wood (2)] is attached at [this point (3)] and [this point (4)] but [it (5)] can rotate. So as [the circle (6)] rotates, [this (7)] moves in and out. So [this whole thing (8)] is just going back and forth."

Even given a high degree of domain knowledge (e.g., that "circles" often "rotate" but "points" rarely do), determining the coreference in this excerpt seems difficult. The word "this" accompanied by a gesture is frequently used to introduce a new entity, so it is difficult to determine from the text alone whether "[this (7)]" refers to "[this piece of wood (2)]," or to an entirely different part of the diagram. In addition, "[this whole thing (8)]" could be anaphoric, or it might refer to a new entity, perhaps some superset of predefined parts.

The example text was drawn from a small corpus of dialogues, which has been annotated for coreference. Participants in the study had little difficulty understanding what was communicated. While this does not prove that human listeners are using gesture or other multimodal features, it suggests that these features merit further investigation. We extracted hand positions from the videos in the corpus, using computer vision. From the raw hand positions, we derived gesture features that were used to supplement traditional textual features for coreference resolution. For a description of the study's protocol, automatic hand tracking, and a fuller examination of the gesture features, see (Eisenstein and Davis, 2006). In this paper, we present results showing that these features yield a significant improvement in performance.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 37–40,
New York, June 2006.

2 Implementation

A set of commonly-used linguistic features were selected for this problem (Table 1). The first five features apply to pairs of NPs; the next set of features are applied individually to both of the NPs that are candidates for coreference. Thus, we include two features each, e.g., **J is PRONOUN** and **I is PRONOUN**, indicating respectively whether the candidate anaphor and candidate antecedent are pronouns. We include separate features for each of the four most common pronouns: "this", "it", "that", and "they," yielding features such as **J="this"**.

2.1 Gesture Features

The gesture features shown in Table 1 are derived from the raw hand positions using a simple, deterministic system. Temporally, all features are computed at the midpoint of each candidate NP; for a further examination of the sensitivity to temporal offset, see (Eisenstein and Davis, 2006).

At most one hand is determined to be the "focus hand," according to the following heuristic: select the hand farthest from the body in the x-dimension, as long as the hand is not occluded and its y-position is not below the speaker's waist. If neither hand meets these criteria, than no hand is said to be in focus. Occluded hands are also not permitted to be in focus; the listener's perspective was very similar to that of the camera, so it seemed unlikely that the speaker would occlude a meaningful gesture. In addition, our system's estimates of the position of an occluded hand are unlikely to be accurate.

If focus hands can be identified during both mentions, the Euclidean distance between focus points is computed. The distance is binned, using the supervised method described in (Fayyad and Irani, 1993). An advantage of binning the continuous features is that we can create a special bin for missing data, which occurs whenever a focus hand cannot be identified.

If the same hand is in focus during both NPs, then the value of **WHICH HAND** is set to "same"; if a different hand is in focus then the value is set to "different"; if a focus hand cannot be identified in one or both NPs, then the value is set to "missing." This multi-valued feature is automatically converted into a set of boolean features, so that all features can be represented as binary variables.

2.2 Coreference Resolution Algorithm

(McCallum and Wellner, 2004) formulates coreference resolution as a Conditional Random Field, where mentions are nodes, and their similarities are represented as weighted edges. Edge weights range from $-\infty$ to ∞, with larger values indicating greater similarity. The optimal solution is obtained by partitioning the graph into cliques such that the sum of the weights on edges within cliques is maximized, and the sum of the weights on edges between cliques is minimized:

$$\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}} \sum_{i,j,i \neq j} y_{i,j} s(x_i, x_j) \tag{1}$$

In equation 1, $\mathbf{x}$ is a set of mentions and $\mathbf{y}$ is a coreference partitioning, such that $y_{i,j} = 1$ if mentions x_i and x_j corefer, and $y_{i,j} = -1$ otherwise. $s(x_i, x_j)$ is a similarity score computed on mentions x_i and x_j.

Computing the optimal partitioning $\hat{\mathbf{y}}$ is equivalent to the problem of correlation clustering, which is known to be NP-hard (Demaine and Immorlica, to appear). Demaine and Immorlica (to appear) propose an approximation using integer programming, which we are currently investigating. However, in this research we use average-link clustering, which hierarchically groups the mentions $\mathbf{x}$, and then forms clusters using a cutoff chosen to maximize the f-measure on the training set.

We experiment with both pipeline and joint models for computing $s(x_i, x_j)$. In the pipeline model, $s(x_i, x_j)$ is the posterior of a classifier trained on pairs of mentions. The advantage of this approach is that any arbitrary classifier can be used; the downside is that minimizing the error on all pairs of mentions may not be equivalent to minimizing the overall error of the induced clustering. For experiments with the pipeline model, we found best results by boosting shallow decision trees, using the Weka implementation (Witten and Frank, 1999).

Our joint model is based on McCallum and Wellner's (2004) adaptation of the voted perceptron to coreference resolution. Here, s is given by the product of a vector of weights λ with a set of boolean features $\phi(x_i, x_j)$ induced from the pair of noun phrases: $s(x_i, x_j) = \lambda\phi(x_i, x_j)$. The maximum likelihood weights can be approximated by a voted perceptron, where, in the iteration t of the perceptron training:

$$\lambda_t = \lambda_{t-1} + \sum_{i,j,i \neq j} \phi(x_i, x_j)(y^*_{i,j} - \hat{y}_{i,j}) \tag{2}$$

In equation 2, $\mathbf{y}^*$ is the ground truth partitioning from the labeled data. $\hat{\mathbf{y}}$ is the partitioning that maximizes equation 1 given the set of weights λ_{t-1}. As before, average-link clustering with an adaptive cutoff is used to partition the graph. The weights are then averaged across all iterations of the perceptron, as in (Collins, 2002).

3 Evaluation

The results of our experiments are computed using mention-based CEAF scoring (Luo, 2005), and are reported in Table 2. Leave-one-out evaluation was used to form 16 cross-validation folds, one for each document in the corpus. Using a planned, one-tailed pairwise t-test, the gesture features improved performance significantly

MARKABLE DIST	The number of markables between the candidate NPs
EXACT MATCH	True if the candidate NPs have identical surface forms
STR MATCH	True if the candidate NPs match after removing articles
NONPRO MATCH	True if the candidate NPs are not pronouns and have identical surface forms
NUMBER MATCH	True if the candidate NPs agree in number
PRONOUN	True if the NP is a pronoun
DEF NP	True if the NP begins with a definite article, e.g. “the box”
DEM NP	True if the NP is not a pronoun and begins with the word “this”
INDEF NP	True if the NP begins an indefinite article, e.g. “a box”
pronouns	Individual features for each of the four most common pronouns: “this”, “it”, “that”, and “they”
FOCUS DIST	Distance between the position of the in-focus hand during j and i (see text)
WHICH HAND	Whether the hand in focus during j is the same as in i (see text)

Table 1: The feature set

System	**Feature set**	**F1**
AdaBoost	Gesture + Speech	54.9
AdaBoost	Speech only	52.8
Voted Perceptron	Gesture + Speech	53.7
Voted Perceptron	Speech only	52.9
Baseline	EXACT MATCH only	50.2
Baseline	None corefer	41.5
Baseline	All corefer	18.8

Table 2: Results

for the boosted decision trees ($t(15) = 2.48, p < .02$), though not for the voted perceptron ($t(15) = 1.07, p = .15$).

In the “all corefer” baseline, all NPs are grouped into a single cluster; in the “none corefer”, each NP gets its own cluster. In the “EXACT MATCH” baseline, two NPs corefer when their surface forms are identical. All experimental systems outperform all baselines by a statistically significant amount. There are few other reported results for coreference resolution on spontaneous, unconstrained speech; (Strube and Müller, 2003) similarly finds low overall scores for pronoun resolution on the Switchboard Corpus, albeit by a different scoring metric. Unfortunately, they do not compare performance to equivalent baselines.

For the AdaBoost method, 50 iterations of boosting are performed on shallow decision trees, with a maximum tree depth of three. For the voted perceptron, 50 training iterations were performed. The performance of the voted perceptron on this task was somewhat unstable, varying depending on the order in which the documents were presented. This may be because a small change in the weights can lead to a very different partitioning, which in turn affects the setting of the weights in the next perceptron iteration. For these results, the order of presentation of the documents was randomized, and the scores for the voted perceptron are the average of 10 different runs ($\sigma = 0.32\%$ with gestures, 0.40% without).

Although the AdaBoost method minimizes pairwise error rather than the overall error of the partitioning, its performance was superior to the voted perceptron. One possible explanation is that by boosting small decision trees, AdaBoost was able to take advantage of non-linear combinations of features. We tested the voted perceptron using all pairwise combinations of features, but this did not improve performance.

4 Discussion

If gesture features play a role in coreference resolution, then one might expect the probability of coreference to vary significantly when conditioned on features describing the gesture. As shown in Table 3, the prediction holds: the binned **FOCUS DIST** gesture feature has the fifth highest χ^2 value, and the relationship between coreference and all gesture features was significant ($\chi^2 = 727.8, dof = 4, p < .01$). Note also that although **FOCUS DIST** ranks fifth, three of the features above it are variants of a string-match feature, and so are highly redundant.

The **WHICH HAND** feature is less strongly correlated with coreference, but the conditional probabilities do correspond with intuition. If the NPs corefer, then the probability of using the same hand to gesture during both NPs is 59.9%; if not, then the likelihood is 52.8%. The probability of not observing a focus hand is 20.3% when the NPs corefer, 25.1% when they do not; in other words, gesture is more likely for both NPs of a coreferent pair than for the NPs of a non-coreferent pair. The relation between the **WHICH HAND** feature and coreference is also significantly different from the null hypothesis ($\chi^2 = 57.2, dof = 2, p < .01$).

Rank	Feature	χ^2
1.	EXACT MATCH	1777.9
2.	NONPRO MATCH	1357.5
3.	STR MATCH	1201.8
4.	J = "it"	732.8
5.	**FOCUS DIST**	727.8
6.	MARKABLE DIST	619.6
7.	J is PRONOUN	457.5
8.	NUMBER	367.9
9.	I = "it"	238.6
10.	I is PRONOUN	132.6
11.	J is INDEF NP	79.3
12.	**SAME FOCUS HAND**	57.2

Table 3: Top 12 Features By Chi-Squared

5 Related Work

Research on multimodality in the NLP community has usually focused on multimodal dialogue systems (e.g., (Oviatt, 1999)). These systems differ fundamentally from ours in that they address human-*computer* interaction, whereas we address human-*human* interaction. Multimodal dialogue systems tackle interesting and difficult challenges, but the grammar, vocabulary, and recognized gestures are often pre-specified, and dialogue is controlled at least in part by the computer. In our data, all of these things are unconstrained.

Prosody has been shown to improve performance on several NLP problems, such as topic and sentence segmentation (e.g., (Shriberg et al., 2000)). We are aware of no equivalent work showing statistically significant improvement on unconstrained speech using hand gesture features. (Nakano et al., 2003) shows that body posture predicts turn boundaries, but does not show that these features improve performance beyond a text-only system. (Chen et al., 2004) shows that gesture may improve sentence segmentation; however, in this study, the improvement afforded by gesture is not statistically significant, and evaluation was performed on a subset of their original corpus that was chosen to include only the three speakers who gestured most frequently. Still, this work provides a valuable starting point for the integration of gesture feature into NLP systems.

6 Conclusion

We have described how gesture features can be used to improve coreference resolution on a corpus of unconstrained speech. Hand position and hand choice correlate significantly with coreference, explaining this gain in performance. We believe this is the first example of hand gesture features improving performance by a statistically significant margin on unconstrained speech.

References

Breck Baldwin and Thomas Morton. 1998. Dynamic coreference-based summarization. In *Proc. of EMNLP*.

Lei Chen, Yang Liu, Mary P. Harper, and Elizabeth Shriberg. 2004. Multimodal model integration for sentence unit detection. In *Proceedings of International Conference on Multimodal Interfaces (ICMI'04)*. ACM Press.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.

Erik D. Demaine and Nicole Immorlica. to appear. Correlation clustering in general weighted graphs. *Theoretical Computer Science*.

Jacob Eisenstein and Randall Davis. 2006. Gesture features for coreference resolution. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms*.

Usama M. Fayyad and Keki B. Irani. 1993. Multi-interval discretization of continuousvalued attributes for classification learning. In *Proceedings of IJCAI-93*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proc. of HLT-EMNLP*, pages 25–32.

Andrew McCallum and Ben Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Neural Information Processing Systems*.

Yukiko Nakano, Gabe Reinstein, Tom Stocky, and Justine Cassell. 2003. Towards a model of face-to-face grounding. In *Proceedings of ACL'03*.

Sharon L. Oviatt. 1999. Mutual disambiguation of recognition errors in a multimodel architecture. In *Human Factors in Computing Systems (CHI'99)*, pages 576–583.

Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tur, and Gokhan Tur. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32.

Michael Strube and Christoph Müller. 2003. A machine learning approach to pronoun resolution in spoken dialogue. In *Proceedings of ACL '03*, pages 168–175.

Ian H. Witten and Eibe Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

Spectral Clustering for Example Based Machine Translation

Rashmi Gangadharaiah
LTI
Carnegie Mellon University
Pittsburgh P.A. 15213
rgangadh@andrew.cmu.edu

Ralf Brown
LTI
Carnegie Mellon University
Pittsburgh P.A. 15213
ralf@cs.cmu.edu

Jaime Carbonell
LTI
Carnegie Mellon University
Pittsburgh P.A. 15213
jgc@cs.cmu.edu

Abstract

Prior work has shown that generalization of data in an Example Based Machine Translation (EBMT) system, reduces the amount of pre-translated text required to achieve a certain level of accuracy (Brown, 2000). Several word clustering algorithms have been suggested to perform these generalizations, such as k-Means clustering or Group Average Clustering. The hypothesis is that better contextual clustering can lead to better translation accuracy with limited training data. In this paper, we use a form of spectral clustering to cluster words, and this is shown to result in as much as 29.08% improvement over the baseline EBMT system.

1 Introduction

In EBMT, the source sentence to be translated is matched against the source language sentences present in a corpus of source-target sentence pairs. When a partial match is found, the corresponding target translations are obtained through subsentential alignment. These partial matches are put together to obtain the final translation by optimizing translation and alignment scores and using a statistical target language model in the decoding process. Prior work has shown that EBMT requires large amounts of data (in the order of two to three million words) (Brown, 2000) of pre-translated text, to function reasonably well. Thus, some modification of the basic EBMT method is required to make it effective when less data is available. In order to use the available text efficiently, systems such as, (Veale and Way, 1997) and (Brown, 1999), convert the examples in the corpus into templates against which the new text can be matched. Thus, source-target sentence pairs are converted to source-target generalized template pairs. An example of such a pair is shown below:

The	session	opened	at	2p.m
La	séance	est ouverte	á	2 heures

The <event> <verb-past-tense> at <time>
La <event> <verb-past-tense> a <time>

This single template can be used to translate different source sentences, including for example,

The	session	adjourned	at	6p.m
The	seminar	opened	at	8a.m

if ‘session’ and ‘seminar’ are both generalized to ‘<event>’, ‘opened’ and ‘adjourned’ are both generalized to ‘<verb-past-tense>’ and finally ‘6p.m’ and ‘8a.m’ are both generalized to ‘<time>’.

The system used by (Brown, 1999) performs its generalization using both equivalence classes of words and a production rule grammar. This paper describes the use of spectral clustering (Ng. et. al., 2001; Zelnik-Manor and Perona, 2004), for automated extraction of equivalence classes. Spectral clustering is seen to be superior to Group Average Clustering (GAC) (Brown, 2000) both in terms of semantic similarity of words falling in a single cluster, and overall BLEU score (Papineni. et. al., 2002) in a large scale EBMT system.

The next section explains the term vectors extracted for each word, which are then used to cluster words into equivalence classes and provides an outline of the Standard GAC algorithm. Section 3 describes the spectral clustering algorithm used. Sec-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 41–44,
New York, June 2006. ©2006 Association for Computational Linguistics

tion 4 lists results obtained in a full evaluation of the algorithm. Section 5 concludes and discusses directions for future work.

2 Term vectors for clustering

Using a bilingual dictionary, usually created using statistical methods such as those of (Brown et. al., 1990) or (Brown, 1997), and the parallel text, a rough mapping between source and target words can be created. This word pair is then treated as an indivisible token for future processing. For each such word pair we then accumulate counts for each token in the surrounding context of its occurrences (N words, currently 3, immediately prior to and N words immediately following). The counts are weighted with respect to distance from occurrence, with a linear decay (from 1 to 1/N) to give greatest importance to the words immediately adjacent to the word pair being examined. These counts form a pseudo-document for each pair, which are then converted into term vectors for clustering.

In this paper, we compare our algorithm against the incremental GAC algorithm(Brown, 2000). This method examines each word pair in turn, computing a similarity measure to every existing cluster. If the best similarity measure is above a predetermined threshold, the new word is placed in the corresponding cluster, otherwise a new cluster is created if the maximum number of clusters has not yet been reached.

3 Spectral clustering

Spectral clustering is a general term used to describe a group of algorithms that cluster points using the eigenvalues of ‘distance matrices’ obtained from data. In our case, the algorithm described in (Ng. et. al., 2001) was performed with certain variations that were proposed by (Zelnik-Manor and Perona, 2004) to compute the scaling factors automatically and for the k-Means orthogonal treatment (Verma and Meila, 2003) during the initialization. These scaling factors help in self-tuning distances between points according to the local statistics of the neighborhoods of the points. The algorithm is briefly described below.

1. Let S =$s_1, s_2, ...s_n$, denote the term vectors to be clustered into k classes.
2. Form the affinity matrix A defined by
   $A_{ij} = exp(-d^2(s_i, s_j)/\sigma_i\sigma_j)$ for $i \neq j$
   $A_{ii} = 1$
   Where, $d(s_i, s_j) = 1/(sim(s_i, s_j) + \epsilon)$
   $sim(s_i, s_j)$ is the Cosine similarity between s_i and s_j, ϵ is used to prevent the ratio from becoming infinity
   σ_i is the set of local scaling parameters for s_i.
   σ_i = $d(s_i, s_T)$ where, s_T is the T^{th} neighbor of point s_i for some fixed T (7 for this paper).
3. Define D to be the diagonal matrix given by,
   $D_{ii} = \Sigma_j A_{ij}$
4. Compute $L = D^{-1/2}AD^{-1/2}$
5. Select k eigenvectors corresponding to k largest eigenvalues (k is presently an externally set parameter). The eigenvectors are normalized to have unit length. Form matrix U by stacking all the eigenvectors in columns.
6. Form the matrix Y by normalizing U’s rows,
   $Y_{ij} = U_{ij}/\sqrt{(\Sigma_j U_{ij}^2)}$
7. Perform k-Means clustering treating each row of Y as a point in k dimensions. The k-Means algorithm is initialized either with random centers or with orthogonal vectors.
8. After clustering, assign the point s_i to cluster c if the corresponding row i of the matrix Y was assigned to cluster c.
9. Sum the distances between the members and the centroid of each cluster to obtain the classification cost.
10. Goto step 7, iterate for a fixed number of iterations. In this paper, 20 iterations were performed with orthogonal k-Means initialization and 5 iterations with random k-Means initialization.
11. The clusters obtained from the iteration with least classification cost are selected as the k clusters.

4 Preliminary Results

The clusters obtained from the spectral clustering method are seen by inspection to correspond to more natural and intuitive word classes than those obtained by GAC. Even though this is subjective and not guaranteed to lead to improve translation performance, it shows that maybe the increased power of spectral clustering to represent non-convex classes (non-convex in the term vector domain) could be useful in a real translation experiment. Some example classes are shown in Table 1. The first class in an intuitive sense corresponds to measurement units. We see that in the <units> case, GAC misses some of the members which are actually distributed among many different classes and hence these are not well generalized. In the second class <months>, spectral clustering has primarily the months in a single class whereas GAC adds a number of seemingly unrelated words to the cluster. The classes were all obtained by finding 80 clusters in a 20,000-sentence pair subset of the IBM Hansard Corpus (Linguistic Data Consortium, 1997) for spectral clustering. 80 was chosen as the number of clusters since it gave the highest BLEU score in the evaluation. For GAC, 300 clusters were used as this gave the best performance.

To show the effectiveness of the clustering methods in an actual evaluation, we set up the following experiment for an English to French translation task on the Hansard corpus. The training data consists of three sets of size 10,000 (set1), 20,000 (set2) and 30,000 (set3) sentence pairs chosen from the first six files of the Hansard Corpus. Only sentences of length 5 to 21 words were taken. Only words with frequency of occurrence greater than 9 were chosen for clustering because more contextual information would be available when the word occurs frequently and this would help in obtaining better clusters. The test data was chosen to be a set of 500 sentences obtained from files 20, 40, 60 and 80 of the Hansard corpus with 125 sentences from each file. Each of the methods was run with different number of clusters and results are reported only for the optimal number of clusters in each case.

The results in Table 2 show that spectral clustering requires moderate amounts of data to get a large improvement. For small amounts of data it is slightly worse than GAC, but neither gives much improvement over the baseline. For larger amounts of data, again both methods are very similar, though spectral clustering is better. Finally, for moderate amounts of data, when generalization is the most useful, spectral clustering gives a significant improvement over the baseline as well as over GAC. By looking at the clusters obtained with varying amounts of data, it can be concluded that high pu-

Table 1: Clusters for <units> and <months>

Spectral clustering	GAC
“adjourned” “hre”	“adjourned” “hre”
“cent” “%”	
“days” “jours”	
“families” “familles”	“families” “familles”
“hours” “heures”	
“million” “millions”	“million” “millions”
“minutes” “minutes”	
“o’clock” “heures”	“o’clock” “heures”
“p.m.” “heures”	“p.m.” “heures”
“p.m.” “hre”	
“people” “personnes”	“people” “personnes”
“per” “%”	“per” “%”
“times” “fois”	“times” “fois”
“years” “ans”	
“august” “août”	“august” “août”
“december” “décembre”	“december” “décembre”
“february” “février”	“february” “février”
“january” “janvier”	“january” “janvier”
“march” “mars”	“march” “mars”
“may” “mai”	“may” “mai”
“november” “novembre”	“november” “novembre”
“october” “octobre”	“october” “octobre”
“only” “seulement”	“only” “seulement”
“june” “juin”	“june” “juin”
“july” “juillet”	“july” “juillet”
“april” “avril”	“april” “avril”
“september” “septembre”	“september” “septembre”
	“page” “page”
	“per” “$”
	“recognize” “parole”
	“recognized” “parole”
	“recorded” “page”
	“section” “article”
“since” “depuis”	“since” “depuis”
	“took” “séance”
	“under” “loi”

Table 2: % Relative improvement over baseline EBMT
# clus is the number of clusters for best performance

	GAC		Spectral	
	% Rel imp	#clus	% Rel imp	#clus
10k	3.33	50	1.37	20
20k	22.47	300	29.08	80
30k	2.88	300	3.88	200

rity clusters can be obtained with even just moderate amounts of data.

5 Conclusions and future work

From the experimental results we see that spectral clustering leads to relatively purer and more intuitive clusters. These clusters result in an improved BLEU score in comparison with the clusters obtained through GAC. GAC can only collect clusters in convex regions in the term vector space, while spectral clustering is not limited in this regard. The ability of spectral clustering to represent non-convex shapes arises due to the projection onto the eigenvectors as described in (Ng. et. al., 2001).

As future work, we would like to analyze the variation in performance as the amount of data increases. It is widely known that increasing the amount of training data in a generalized EBMT system eventually leads to saturation of performance, where all clustering methods perform about as well as baseline. Thus, all methods have an operating region where they are the most useful. We would like to locate and extend this region for spectral clustering.

Also, it would be interesting to compare the clusters obtained with spectral clustering and the Part of Speech tags of the words in the same cluster, especially for languages such as English where good taggers are available.

Finally, an important direction of research is in automatically selecting the number of clusters for the clustering algorithm. To do this, we could use information from the eigenvalues or the distribution of points in the clusters.

Acknowledgment

This work was funded by National Business Center award NBCHC050082.

References

Andrew Ng, Michael Jordan, and Yair Weiss 2001. On Spectral Clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference,* pages 849-856, Vancouver, British Columbia, Canada, December.

Deepak Verma and Marina Meila. 2003. Comparison of Spectral Clustering Algorithms. `http://www.ms.washington.edu/~spectral/`.

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 of the Association for Computational Linguistics (ACL 2002),* pages 311-318,Philadelphia, PA, July. `http://acl.ldc.upenn.edu/P/P02`

Linguistic Data Consortium. 1997. *Hansard Corpus of Parallel English and French.* Linguistic Data Consortium, December. `http://www.ldc.upenn.edu/`

L. Zelnik-Manor and P. Perona 2004 Self-Tuning Spectral Clustering. In *Advances in Neural Information Processing Systems 17: Proceeding of the 2004 Conference.*

Peter Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer and P. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics,* 16:79-85.

Ralf D. Brown. 1997. Automated Dictionary Extraction for “Knowledge-Free” Example-Based Translation. In *Proceedings of the Seventh International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-97),* pages 111-118, Santa Fe, New Mexico, July. `http://www.cs.cmu.edu/~ralf/papers.html`

Ralf D. Brown. 1999. Adding Linguistic Knowledge to a Lexical Example-Based Translation System. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation(TMI-99),* pages 22-32, August. `http://www.cs.cmu.edu/~ralf/papers.html`

Ralf. D. Brown. 2000. Automated Generalization of Translation Examples. In *Proceedings of Eighteenth International Conference on Computational Linguistics (COLING-2000),* pages 125-131, Saarbrücken, Germany.

Tony Veale and Andy Way. 1997. Gaijin: A Template-Driven Bootstrapping Approach to Example-Based Machine Translation. In *Proceedings of NeMNLP97, New Methods in Natural Language Processing,* Sofia, Bulgaria, September. `http://www.compapp.dcu.ie/~tonyv/papers/gaijin.html`.

A Finite-State Model of Georgian Verbal Morphology

Olga Gurevich
Department of Linguistics
University of California, Berkeley
3834 23rd Street
San Francisco, CA 94114
olya.gurevich@gmail.com

Abstract

Georgian is a less commonly studied language with complex, non-concatenative verbal morphology. We present a computational model for generation and recognition of Georgian verb conjugations, relying on the analysis of Georgian verb structure as a word-level template. The model combines a set of finite-state transducers with a default inheritance mechanism.[1]

1 Introduction

Georgian morphology is largely synthetic, with complex verb forms that can often express the meaning of a whole sentence. Descriptions of Georgian verbal morphology emphasize the large number of inflectional categories; the large number of elements that a verb form can contain; the inter-dependencies in the occurrence of various elements; and the large number of regular, semi-regular, and irregular patterns of formation of verb inflections (cf. Hewitt 1995). All of these factors make computational modeling of Georgian morphology a rather daunting task.

In this paper, we propose a computational model for parsing and generation of a subset of Georgian verbs that relies on a templatic, word-based analysis of the verbal system rather than assuming compositional rules for combining individual morphemes. We argue that such a model is viable, extensible, and capable of capturing the generalizations inherent in the Georgian verbal system at various levels of regularity. To our knowledge, this is the only computational model of the Georgian verb currently in active development and available to the non-Georgian academic community[2].

2 Georgian Verbal Morphology

The Georgian verb forms are made up of several kinds of morphological elements that recur in different formations. These elements can be formally identified in a fairly straightforward fashion; however, their function and distribution defy a simple compositional analysis but instead are determined by the larger morphosyntactic and semantic contexts in which the verbs appear (usually tense, aspect, and mood) and the lexical properties of the verbs themselves.

2.1 Verb Structure

Georgian verbs are often divided into four conjugation classes, based mostly on valency (cf. Harris 1981). In this brief report, we will concentrate on transitive verbs, although our model can accommodate all four conjugation types. Verbs inflect in tense/mood/aspect (TAM) paradigms (simplified here as tenses). There are a total of 10 actively used tenses in Modern Georgian, grouped into TAM series as in Table 1. Knowing the series and tense of a verb form is essential for being able to conjugate it.

The structure of the verb can be described using the following (simplified) template.

[1] This work was in part supported by the Berkeley Language Center. I'd like to thank Lauri Karttunen for introducing me to finite-state morphology and providing an updated version of the software, and Shorena Kurtsikidze and Vakhtang Chikovani for help with the Georgian data. All errors are my own.

[2] See Tandashvili (1999) for an earlier model. Unfortunately, the information in the available publications does not allow for a meaningful comparison with the present model.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 45–48,
New York, June 2006.

Series	Tense	2SGSUBJ:3SGOBJ
I	PRESENT	***xat’***-*av*
	IMPERFECT	***xat’***-*av-di*
	PRES. SUBJ.	***xat’***-*av-de*
	FUTURE	*da*-***xat’***-*av*
	CONDITIONAL	*da*-***xat’***-*av-di*
	FUT. SUBJ.	*da*-***xat’***-*av-de*
II	AORIST	*da*-***xat’***-*e*
	AOR. SUBJ.	*da*-***xat’***-*o*
III	PERFECT	*da-gi*-***xat’***-*av-s*
	PLUPERFECT	*da-ge*-***xat’***-*a*

Table 1: Tenses of the verb ‘to paint’. Root is in bold.

(Preverb)-(agreement1)-(version)-**root**-(thematic suffix)-(tense)-(agreement)

The functions of some of the elements are discussed below. As an illustration, note the formation of the verb xat’va ‘paint’ in Table 1.

2.2 Lexical and Semi-Regular Patterns

The complexity of the distribution of morphological elements in Georgian is illustrated by preverbs, thematic suffixes, and tense endings. The preverbs (a closed class of about 8) indicate perfective aspect and lexical derivations from roots, similar to verb prefixes in Slavic or German. The association of a verb with a particular preverb is lexical and must be memorized. A preverb appears on forms from the Future subgroup of series I, and on all forms of series II and III in transitive verbs. Table 2 demonstrates some of the lexically-dependent morphological elements, including several different preverbs (row ‘Future’).

Similarly, thematic suffixes form a closed class and are lexically associated with verb roots. They function as stem formants and distinguish inflectional classes. In transitive verbs, thematic suffixes appear in all series I forms. Their behavior in other series differs by individual suffix: in series II, most suffixes disappear, though some seem to leave partial “traces” (rows ‘Present’ and ‘Perfect’ in Table 2).

The next source of semi-regular patterns comes from the inflectional endings in the individual tenses and the corresponding changes in some verb roots (row ‘Aorist’ in Table 2).

Finally, another verb form relevant for learners is the masdar, or verbal noun. The masdar may or may not include the preverb and/or some variation of the thematic suffix (last row in Table 2).

	‘Bring’	‘Paint’	‘Eat’
Present	i-**gh**-*eb*-s	**xat’**-*av*-s	**ch’am**-*ø*-s
Future	*c’amo*-i-**gh**-eb-s	*da*-**xat’**-av-s	*she*-**ch’am**-s
Aorist	c’amo-i-**gh**-*o*	da-**xat’**-*a*	she-**ch’am**-*a*
Perfect	c’amo-u-**gh**-*ia*	da-u-**xat’**-*av-s*	she-u-**ch’am**-*ia*
Masdar	c’amo-**gh**-eb-a	da-**xat’**-v-a	**ch’**-am-a

Table 2: Lexical Variation. Roots are in bold; lexically variable affixes are in italics.

SUBJ	OBJ				
	1SG	1PL	2SG	2PL	3
1SG	—	—	g—ø	g—t	v—ø
1PL	—	—	g—t	g—t	v—t
2SG	m—ø	gv—ø	—	—	ø—ø
2PL	m—t	gv—t	—	—	—t
3SG	m—*	gv—*	g—*	g—t	—*
3PL	m—**	gv—**	g—**	g—**	—**

Table 3: Subject/Object agreement. The 3sg and 3pl suffixes, marked by * and **, are tense-dependent.

2.3 Regular Patterns

Verb agreement in Georgian is a completely regular yet not entirely compositional phenomenon. A verb can mark agreement with both the subject and the object via a combination of prefixal and suffixal agreement markers, as in Table 3.

The distribution and order of attachment of agreement affixes has been the subject of much discussion in theoretical morphological literature. To simplify matters for the computational model, we assume here that the prefixal and suffixal markers attach to the verb stem at the same time, as a sort of circumfix, and indicate the combined subject and object properties of a paradigm cell.

Despite the amount of lexical variation, tense formation in some instances is also quite regular. So, the Imperfect and First Subjunctive tenses are regularly formed from the Present. Similarly, the Conditional and Future Subjunctive are formed from the Future. And for most (though not all) transitive verbs, the Future is formed from the Present via the addition of a preverb.

Additionally, the number of possible combinations of inflectional endings and other irregularities is also finite, and some choices tend to predict other choices in the paradigm of a given verb. Georgian verbs can be classified according to several example paradigms, or inflectional (lexical) classes, similar to the distinctions made in Standard European languages; the major difference is that the number of classes is much greater in Georgian. For instance, Melikishvili (2001) distinguishes over 60 classes, of which 17 are transitive. While the exact number of inflectional classes is still in question, the general example-based approach seems the only one viable for Georgian.

3 Computational Model

3.1 Overview

Finite-state networks are currently one of the most popular methods in computational morphology. Many approaches are implemented as two-way finite-state transducers (FST) in which each arc corresponds to a mapping of two elements, for example a phoneme and its phonetic realization or a morpheme and its meaning. As a result, FST morphologies often assume morpheme-level compositionality. As demonstrated in the previous section, such assumptions do not serve well to describe the verbal morphology of Georgian. Instead, it can be described as a series of patterns at various levels of regularity. However, compositionality is not a necessary assumption: finite-state models are well-suited for representing mappings from strings of meaning elements to strings of form elements without necessarily pairing them one-to-one.

Our model was implemented using the *xfst* program included in (Beesley and Karttunen 2003). The core of the model consists of several levels of finite-state transducer (FST) networks such that the result of compiling a lower-level network serves as input to a higher-level network. The levels correspond to the division of templatic patterns into completely lexical (Level 1) and semi-regular (Level 2). Level 3 contains completely regular patterns that apply to the results of both Level 1 and Level 2. The regular-expression patterns at each level are essentially constraints on the templatic structure of verb forms at various levels of generality. The FST model can be used both for the generation of verbal inflections and for recognition of complete forms.

The input to the model is a set of hand-written regular expressions (written as FST patterns) which identify the lexically specific information for a representative of each verb class, as well as the more regular rules of tense formation. In addition to dividing verb formation patterns into lexical and regular, our model also provides a mechanism for specifying defaults and overrides in inflectional markers. Many of the tense-formation patterns mentioned above can be described as defaults with some lexical exceptions. In order to minimize the amount of manual entry, we specify the exceptional features at the first level and use the later levels to apply default rules in all other cases.

3.2 Level 1: The Lexicon

The first level of the FST model contains lexically specific information stored as several complete word forms for each verb. In addition to the information that is always lexical (such as the root and preverb), this network also contains forms which are exceptional. For the most regular verbs, these are: Present, Future, Aorist 2SgSubj, Aorist 3SgSubj, and Perfect.

The inflected forms are represented as two-level finite-state arcs, with the verb stem and morphosyntactic properties on the upper side, and the inflected word on the lower side.

The forms at Level 1 contain a place holder "+Agr1" for the prefixal agreement marker, which is replaced by the appropriate marker in the later levels (necessary because the prefixal agreement is between the preverb and the root).

3.3 Level 2: Semi-regular Patterns

The purpose of Level 2 is to compile inflectional forms that are dependent on other forms (introduced in Level 1), and to provide default inflections for regular tense formation patterns.

An example of the first case is the Conditional tense, formed predictably from the Future tense. The FST algorithm is as follows:

- Compile a network consisting of Future forms.
- Add the appropriate inflectional suffixes.
- Replace the tense property "+Fut" with "+Cond".
- Add the inflectional properties where needed.

An example of the second case is the Present 3PlSubj suffix, which is *-en* for most transitive verbs, but *-ian* for a few others (see Fig. 1). Xfst provides a simplified feature unification mechanism called *flag*

Lev. 1	*paint*+Pres *xat’-**av***	*paint*+Aor ***da**-xat’-**a***	*open*+PresPl *xsn-**ian***
Lev. 2	*paint*+Past+3Sg *xat’-av-**da***	*paint*+Pres+3Pl *xat’-av-**en***	default overridden
Lev. 3	*paint*+3PlSubj+1SgObj ***m**-xat’-av-en*	*open*+3PlSubj+1SgObj ***m**-xsn-ian*	

Figure 1: Verbs ‘paint’ and ‘open’ at three levels of the model. New information contributed by each form is in bold.

diacritics. Using these flags, we specify exceptional forms in Level 1, so that default inflections do not apply to them in Level 2.

The patterns defined at Level 2 are compiled into a single network, which serves as input to Level 3.

3.4 Level 3: Regular Patterns

The purpose of Level 3 is to affix regular inflection: object and non-3rd person subject agreement. As described in section 2, agreement in Georgian is expressed via a combination of a pre-stem affix and a suffix, which are best thought of as attaching simultaneously and working in tandem to express both subject and object agreement. Thus the compilation of Level 3 consists of several steps, each of which corresponds to a paradigm cell.

The operation of the model is partially illustrated on forms of the verbs ‘paint’ and ‘open’ in Figure 1.

3.5 Treatment of Lexical Classes

The input to Level 1 contains a representative for each lexical class, supplied with a diacritic feature indicating the class number. Other verbs that belong to those classes could, in principle, be inputted along with the class number, and the FST model could substitute the appropriate roots in the process of compiling the networks. However, there are several challenges to this straightforward implementation. Verbs belonging to the same class may have different preverbs, thus complicating the substitution. For many verbs, tense formation involves stem alternations such as syncope or vowel epenthesis, again complicating straightforward substitution. Suppletion is also quite common in Georgian, requiring completely different stems for different tenses.

As a result, even for a verb whose lexical class is known, several pieces of information must be supplied to infer the complete inflectional paradigm. The FST substitution mechanisms are fairly restricted, and so the compilation of new verbs is done in Java. The scripts make non-example verbs look like example verbs in Level 1 of the FST network by creating the necessary inflected forms, but the human input to the scripts need only include the information necessary to identify the lexical class of the verb.

4 Evaluation and Future Work

At the initial stages of modeling, we have concentrated on regular transitive verbs and frequent irregular verbs. The model currently contains several verbs from each of the 17 transitive verb classes mentioned in (Melikishvili 2001), and a growing number of frequent irregular verbs from different conjugation classes. Regular unaccusative, unergative, and indirect verbs will be added in the near future, with the goal of providing full inflections for 200 most frequent Georgian verbs.

The model serves as the basis for an online learner’s reference for Georgian conjugations (Gurevich 2005), which is the only such reference currently available.

A drawback of most finite-state models is their inability to generalize to novel items the way a human could. However, the output of our finite-state model could potentially be used to generate training sets for connectionist or statistical models.

References

Beesley, Kenneth and Lauri Karttunen. 2003. *Finite-State Morphology*. Cambridge University Press.

Gurevich, Olga. 2005. Computing non-concatenative morphology: The case of georgian. In *LULCL 2006*. Bolzano, Italy.

Harris, Alice C. 1981. *Georgian syntax: a study in relational grammar*. Cambridge University Press.

Hewitt, B. G. 1995. *Georgian: a structural reference grammar*. John Benjamins.

Melikishvili, Damana. 2001. *Kartuli zmnis ughlebis sist’ema [Conjugation system of the Georgian verb]*. Logos presi.

Tandashvili, M. 1999. *Main Principles of Computer-Aided Modeling, http://titus.uni-frankfurt.de/personal/manana/refeng.htm*. Tbilisi Habilitation.

Arabic Preprocessing Schemes for Statistical Machine Translation

Nizar Habash
Center for Computational Learning Systems
Columbia University
habash@cs.columbia.edu

Fatiha Sadat
Institute for Information Technology
National Research Council of Canada
fatiha.sadat@cnrc-nrc.gc.ca

Abstract

In this paper, we study the effect of different word-level preprocessing decisions for Arabic on SMT quality. Our results show that given large amounts of training data, splitting off only proclitics performs best. However, for small amounts of training data, it is best to apply English-like tokenization using part-of-speech tags, and sophisticated morphological analysis and disambiguation. Moreover, choosing the appropriate preprocessing produces a significant increase in BLEU score if there is a change in genre between training and test data.

1 Introduction

Approaches to statistical machine translation (SMT) are robust when it comes to the choice of their input representation: the only requirement is consistency between training and evaluation.[1] This leaves a wide range of possible preprocessing choices, even more so for morphologically rich languages such as Arabic. We use the term "preprocessing" to describe various input modifications that can be applied to raw training and evaluation texts for SMT to make them suitable for model training and decoding, including different kinds of tokenization, stemming, part-of-speech (POS) tagging and lemmatization. We refer to a specific kind of preprocessing as a "scheme" and differentiate it from the "technique" used to obtain it. Since we wish to study the effect of word-level preprocessing, we do not utilize any syntactic information. We define the word (and by extension its morphology) to be limited to written Modern Standard Arabic (MSA) strings separated by white space, punctuation and numbers. Thus, some prepositional particles and conjunctions are considered part of the word morphology.

In this paper, we report on an extensive study of the effect on SMT quality of six preprocessing schemes[2], applied to text disambiguated in three different techniques and across a learning curve. Our results are as follows: (a) for large amounts of training data, splitting off only proclitics performs best; (b) for small amount of training data, following an English-like tokenization and using part-of-speech tags performs best; (c) suitable choice of preprocessing yields a significant increase in BLEU score if there is little training data and/or there is a change in genre between training and test data; (d) sophisticated morphological analysis and disambiguation help significantly in the absence of large amounts of data.

Section 2 presents previous relevant research. Section 3 presents some relevant background on Arabic linguistics to motivate the schemes discussed in Section 4. Section 5 presents the tools and data sets used, along with the results of our experiments. Section 6 contains a discussion of the results.

2 Previous Work

The anecdotal intuition in the field is that reduction of word sparsity often improves translation quality. This reduction can be achieved by increasing training data or via morphologically driven preprocessing (Goldwater and McClosky, 2005). Recent publications on the effect of morphology on SMT quality focused on morphologically rich languages such as German (Nießen and Ney, 2004); Spanish, Catalan, and Serbian (Popović and Ney, 2004); and Czech (Goldwater and McClosky, 2005). They all studied

[1] This paper is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA. We thank Roland Kuhn, George Forster, Mona Diab, Owen Rambow, and Martin Jansche for helpful discussions.

[2] We conducted several additional experiments that we do not report on here for lack of space but we reserve for a separate technical report.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 49–52,
New York, June 2006. ©2006 Association for Computational Linguistics

the effects of various kinds of tokenization, lemmatization and POS tagging and show a positive effect on SMT quality. Specifically considering Arabic, Lee (2004) investigated the use of automatic alignment of POS tagged English and affix-stem segmented Arabic to determine appropriate tokenizations. Her results show that morphological preprocessing helps, but only for the smaller corpora. As size increases, the benefits diminish. Our results are comparable to hers in terms of BLEU score and consistent in terms of conclusions. We extend on previous work by experimenting with a wider range of preprocessing schemes for Arabic, by studying the effect of morphological disambiguation (beyond POS tagging) on preprocessing schemes over learning curves, and by investigating the effect on different genres.

3 Arabic Linguistic Issues

Arabic is a morphologically complex language with a large set of morphological features. These features are realized using both concatenative (affixes and stems) and templatic (root and patterns) morphology with a variety of morphological and phonological adjustments that appear in word orthography and interact with orthographic variations. Certain letters in Arabic script are often spelled inconsistently which leads to an increase in both sparsity (multiple forms of the same word) and ambiguity (same form corresponding to multiple words). For example, variants of Hamzated Alif, أ or إ are often written without their Hamza (ء): ا. Another example is the optionality of diacritics in Arabic script. We assume all of the text we are using is undiacritized.

Arabic has a set of attachable clitics to be distinguished from inflectional features such as gender, number, person and voice. These clitics are written attached to the word and thus increase its ambiguity. We can classify three degrees of cliticization that are applicable in a strict order to a word base:

`[CONJ+ [PART+ [Al+ BASE +PRON]]]`

At the deepest level, the BASE can have a definite article (Al+ *the*)[3] or a member of the class of pronominal enclitics, +PRON, (e.g. +hm *their/them*). Next comes the class of particle proclitics (PART+): l+ *to/for*, b+ *by/with*, k+ *as/such* and s+ *will/future*. Most shallow is the class of conjunction proclitics (CONJ+): w+ *and* and f+ *then*.

These phenomena highlight two issues related to preprocessing: First, ambiguity in Arabic words is an important issue to address. To determine whether a clitic or feature should be split off or abstracted off requires that we determine that said feature is indeed present in the word we are considering in context – not just that it is possible given an analyzer or, worse, because of regular expression matching. Secondly, once a specific analysis is determined, the process of splitting off or abstracting off a feature must be clear on what the form of the resulting word is to be. For example, the word كتبتهم *ktbthm* has two possible readings (among others) as *their writers* or *I wrote them*. Splitting off the pronominal clitic +*hm* without normalizing the *t* to *p* in the nominal reading leads to the coexistence of two forms of the noun: *ktbp* and *ktbt*. This increased sparsity is only worsened by the fact that the second form is also the verbal form (thus increased ambiguity).

4 Preprocessing: Schemes and Techniques

A scheme is a specification of the form of preprocessed output; whereas a technique is the method used to create such output. We examine six different schemes and three techniques.

4.1 Preprocessing Techniques

The different techniques chosen illustrate three degrees of linguistic knowledge dependence. The first is very light and cheap. The second is more expensive, requiring the use of a morphological analyzer. And the third is yet more expensive than the second; it is a disambiguation system that requires an analyzer and a disambiguated training corpus.

- REGEX is the baseline technique. It is simply greedy regular expression matching to modify strings and/or split off prefix/suffix substrings that look like clitics indicated by specific schemes. REGEX cannot be used with complex schemes such as EN and MR (see Section 4.2).
- BAMA, Buckwalter Arabic Morphological Analyzer (Buckwalter, 2002), is used to obtain possible word analyses. Using BAMA prevents incorrect greedy REGEX matches. Since BAMA produces multiple analyses, we always select one in a consistent arbitrary manner (first in a sorted list of analyses).
- MADA, The Morphological Analysis and Disambiguation for Arabic tool, is an off-the-shelf resource for Arabic disambiguation (Habash and

[3] Arabic transliterations are provided in the Buckwalter transliteration scheme (Buckwalter, 2002).

Table 1: The Different Preprocessing Schemes (with **MADA** Technique)

Input	wsynhY	Alr}ys	jwlth	bzyArp	AlY	trkyA.	
Gloss	and will finish	the president	tour his	with visit	to	Turkey	.
English	The president will finish his tour with a visit to Turkey.						
ST	wsynhY	Alr}ys	jwlth	bzyArp	AlY	trkyA	.
D1	w+ synhy	Alr}ys	jwlth	bzyArp	<lY	trkyA	.
D2	w+ s+ ynhy	Alr}ys	jwlth	b+ zyArp	<lY	trkyA	.
D3	w+ s+ ynhy	Al+ r}ys	jwlp +P$_{3MS}$	b+ zyArp	<lY	trkyA	.
MR	w+ s+ y+ nhy	Al+ r}ys	jwl +p +h	b+ zyAr +p	<lY	trkyA	.
EN	w+ s+ >nhY$_{VBP}$ +S$_{3MS}$	Al+ r}ys$_{NN}$	jwlp$_{NN}$ +P$_{3MS}$	b+ zyArp$_{NN}$	<lY$_{IN}$	trkyA$_{NNP}$	.

Rambow, 2005). MADA selects among BAMA analyses using a combination of classifiers for 10 orthogonal dimensions, including POS, number, gender, and pronominal clitics.

For BAMA and MADA, applying a preprocessing scheme involves moving features (as specified by the scheme) out of the chosen word analysis and regenerating the word without the split off features (Habash, 2004). The regeneration guarantees the normalization of the word form.

4.2 Preprocessing Schemes

Table 1 exemplifies the effect of the different schemes on the same sentence.

• **ST**: Simple Tokenization is the baseline preprocessing scheme. It is limited to splitting off punctuations and numbers from words and removing any diacritics that appear in the input. This scheme requires no disambiguation.

• **D1, D2, and D3**: Decliticizations. D1 splits off the class of conjunction clitics (*w*+ and *f*+). D2 splits off the class of particles (*l*+, *k*+, *b*+ and *s*+) beyond D1. Finally D3 splits off what D2 does in addition to the definite article (*Al*+) and all pronominal clitics.

• **MR**: Morphemes. This scheme breaks up words into stem and affixival morphemes.

• **EN**: English-like. This scheme is intended to minimize differences between Arabic and English. It decliticizes similarly to D3; however, it uses lexeme and English-like POS tags instead of the regenerated word and it indicates the pro-dropped verb subject explicitly as a separate token.

5 Experiments

We use the phrase-based SMT system, Portage (Sadat et al., 2005). For training, Portage uses IBM word alignment models (models 1 and 2) trained in both directions to extract phrase tables. Maximum phrase size used is 8. Trigram language models are implemented using the SRILM toolkit (Stolcke, 2002). Decoding weights are optimized using Och's algorithm (Och, 2003) to set weights for the four components of the log-linear model: language model, phrase translation model, distortion model, and word-length feature. The weights are optimized over the BLEU metric (Papineni et al., 2001). The Portage decoder, Canoe, is a dynamic-programming beam search algorithm, resembling the algorithm described in (Koehn, 2004a).

All of the training data we use is available from the Linguistic Data Consortium (LDC). We use an Arabic-English parallel corpus of about 5 million words for translation model training data.[4] We created the English language model from the English side of the parallel corpus together with 116 million words from the English Gigaword Corpus (LDC2005T12) and 128 million words from the English side of the UN Parallel corpus (LDC2004E13). English preprocessing comprised down-casing, separating punctuation from words and splitting off "'s". Arabic preprocessing was varied using the proposed schemes and techniques. Decoding weight optimization was done on 200 sentences from the 2003 NIST MT evaluation test set. We used two different test sets: (a) the 2004 NIST MT evaluation test set (MT04) and (b) the 2005 NIST MT evaluation test set (MT05). MT04 is a mix of news, editorials and speeches, whereas MT05, like the training data, is purely news. We use the evaluation metric BLEU-4 (Papineni et al., 2001).

We conducted all possible combinations of schemes and techniques discussed in Section 4 with different training corpus sizes: 1%, 10% and 100%. The results of the experiments are summarized in

[4] The parallel text includes Arabic News, eTIRR, English translation of Arabic Treebank, and Ummah.

Table 2: Results

	MT04									MT05								
	MADA			BAMA			REGEX			MADA			BAMA			REGEX		
	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100
ST	9.4	22.9	34.6	9.4	22.9	34.6	9.4	22.9	34.6	11.2	27.7	37.8	11.2	27.7	**37.8**	11.2	27.7	37.8
D1	13.1	26.9	36.1	12.9	26.5	35.6	11.4	25.5	34.8	14.9	29.8	37.3	14.5	29.6	37.0	13.2	29.5	38.5
D2	14.2	27.7	**37.1**	13.7	27.9	**36.2**	12.0	25.5	**35.8**	16.3	30.2	**38.6**	15.5	31.0	**37.8**	13.4	**29.8**	**38.7**
D3	16.5	**28.7**	34.3	15.9	**28.3**	34.2	**13.6**	**26.1**	34.0	17.7	**31.0**	36.0	17.3	**31.1**	35.3	**14.7**	28.8	36.1
MR	11.6	27.5	34.4	14.2	27.5	33.4	n/a	n/a	n/a	12.7	29.6	35.9	15.7	29.5	34.3	n/a	n/a	n/a
EN	**17.5**	28.4	34.5	**16.3**	27.9	34.0	n/a	n/a	n/a	**18.3**	30.4	36.0	**17.6**	30.4	34.8	n/a	n/a	n/a

Table 2. All reported scores must have over 1.1% BLEU-4 difference to be significant at the 95% confidence level for 1% training. For all other training sizes, the difference must be over 1.7% BLEU-4. Error intervals were computed using bootstrap resampling (Koehn, 2004b).

6 Discussion

Across different schemes, **EN** performs the best under scarce-resource condition; and **D2** performs best under large-resource condition. Across techniques and under scarce-resource conditions, MADA is better than BAMA which is better than REGEX. Under large-resource conditions, this difference between techniques is statistically insignificant, though it's generally sustained across schemes.

The baseline for MT05, which is fully in news genre like training data, is considerably higher than MT04 (mix of genres). To investigate the effect of different schemes and techniques on different genres, we isolated in MT04 those sentences that come from the editorial and speech genres. We performed similar experiments as reported above on this subset of MT04. We found that the effect of the choice of the preprocessing technique+scheme was amplified. For example, MADA+**D2** (with 100% training) on non-news improved the system score 12% over the baseline **ST** (statistically significant) as compared to 2.4% for news only.

Further analysis shows that combination of output from all six schemes has a large *potential* improvement over all of the different systems, suggesting a high degree of complementarity. For example, a 19% improvement in BLEU score (for MT04 under MADA with 100% training) (from 37.1 in **D2** to 44.3) was found from an oracle combination created by selecting for each input sentence the output with the highest sentence-level BLEU score.

7 Future Work

We plan to study additional variants that these results suggest may be helpful. In particular, we plan to include more syntactic knowledge and investigate combination techniques at the sentence and subsentence levels.

References

T. Buckwalter. 2002. Buckwalter Arabic Morphological Analyzer. Linguistic Data Consortium. (LDC2002L49).

S. Goldwater and D. McClosky. 2005. Improving Statistical MT through Morphological Analysis. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

N. Habash. 2004. Large Scale Lexeme Based Arabic Morphological Generation. In *Proc. of Traitement Automatique du Langage Naturel*.

N. Habash and O. Rambow. 2005. Tokenization, Morphological Analysis, and Part-of-Speech Tagging for Arabic in One Fell Swoop. In *Proc. of the Association for Computational Linguistics (ACL)*.

P. Koehn. 2004a. Pharaoh: a Beam Search Decoder for Phrase-based Statistical Machine Translation Models. In *Proc. of the Association for Machine Translation in the Americas*.

P. Koehn. 2004b. Statistical Significance Tests For Machine Translation Evaluation. In *Proc. of EMNLP*.

Y. Lee. 2004. Morphological Analysis for Statistical Machine Translation. In *Proc. of the North American Chapter of ACL*.

F. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176(W0109-022), IBM Research.

M. Popović and H. Ney. 2004. Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. In *Proc. of the Conference on Language Resources and Evaluation*.

S. Nießen and H. Ney. 2004. Statistical Machine Translation with Scarce Resources Using Morpho-syntactic Information. *Computational Linguistics*, 30(2).

F. Sadat, H. Johnson, A. Agbago, G. Foster, R. Kuhn, J. Martin, and A. Tikuisis. 2005. Portage: A Phrase-based Machine Translation System. In *Proc. of ACL Workshop on Building and Using Parallel Texts*.

Andreas Stolcke. 2002. Srilm - An Extensible Language Modeling Toolkit. In *Proc. of International Conference on Spoken Language Processing*.

Agreement/Disagreement Classification: Exploiting Unlabeled Data using Contrast Classifiers

Sangyun Hahn **Richard Ladner**
Dept. of Computer Science and Engineering
University of Washington, Seattle, WA
{syhahn,ladner}@cs.washington.edu

Mari Ostendorf
Dept. of Electrical Engineering
University of Washington, Seattle, WA
mo@ee.washington.edu

Abstract

Several semi-supervised learning methods have been proposed to leverage unlabeled data, but imbalanced class distributions in the data set can hurt the performance of most algorithms. In this paper, we adapt the new approach of contrast classifiers for semi-supervised learning. This enables us to exploit large amounts of unlabeled data with a skewed distribution. In experiments on a speech act (agreement/disagreement) classification problem, we achieve better results than other semi-supervised methods. We also obtain performance comparable to the best results reported so far on this task and outperform systems with equivalent feature sets.

1 Introduction

In natural language understanding research with data-driven techniques, data labeling is an essential but time-consuming and costly process. To alleviate this effort, various semi-supervised learning algorithms such as self-training (Yarowsky, 1995), co-training (Blum and Mitchell, 1998; Goldman and Zhou, 2000), transductive SVM (Joachims, 1999) and many others have been proposed and successfully applied under different assumptions and settings. They all aim to improve classification accuracy by exploiting more readily available unlabeled data as well as labeled examples. However, these iterative training methods have shortcomings when trained on data with imbalanced class distributions. One reason is that most classifiers underlying these methods assume a balanced training set, and thus when one of the classes has a much larger number of examples than the other classes, the trained classifier will be biased toward the majority class. The imbalance will propagate through subsequent iterations, resulting in a more skewed data set upon which a further biased classifier will be trained. To exploit unlabeled data in learning an inherently skewed data distribution, we introduce a semi-supervised classification method using contrast classifiers, first proposed by Peng *et al.* (Peng et al., 2003). It approximates the posterior class probability given an observation using class-specific contrast classifiers that implicitly model the difference between the distribution of labeled data for that class and the unlabeled data.

In this paper, we will explore the applicability of contrast classifiers to the problem of semi-supervised learning for identifying agreements and disagreements in multi-party conversational speech. These labels represent a simple type of "speech act" that can be important for understanding the interaction between speakers, or for automatically summarizing or browsing the contents of a meeting. This problem was previously studied (Hillard et al., 2003; Galley et al., 2004), using a subset of ICSI meeting recording corpus (Janin et al., 2003). In semi-supervised learning, there is a challenge due to an imbalanced class distribution: over 60% of the data are associated with the default class and only 5% are with disagreements.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 53–56,
New York, June 2006. ©2006 Association for Computational Linguistics

2 Contrast Classifier

The contrast classifier approach was developed by Peng *et al* and successfully applied to the problem of identifying protein disorder in a protein structure database (outlier detection) and to finding articles about them (single-class detection) (Peng et al., 2003). A contrast classifier discriminates between the labeled and unlabeled data, and can be used to approximate the posterior class probability of a given data instance as follows. Taking a Bayesian approach, a contrast classifier for the j-th class is defined as:

$$cc_j(x) = \frac{r_j g(x)}{(1 - r_j)h_j(x) + r_j g(x)} \tag{1}$$

where $h_j(x)$ is the likelihood of x generated by class j in the labeled data, $g(x)$ is the distribution of unlabeled data, and r_j is the relative proportion of unlabeled data compared to the labeled data for class j. This discriminates the class j in the labeled data from the unlabeled data. Here, we constrain $r_j = 0.5$ for all j, using resampling to address class distribution skew, as described below. Rewriting equation 1, $h_j(x)$ can be expressed in terms of $cc_j(x)$ as:

$$h_j(x) = \frac{1 - cc_j(x)}{cc_j(x)} \cdot \frac{r}{1 - r} \cdot g(x). \tag{2}$$

Then, the posterior probability of an input x for class j, $p(j|x)$, can be approximated as:

$$p(j|x) = \frac{h_j(x)q_j}{\sum_i h_i(x)q_i} \tag{3}$$

where q_j is the prior class probability which can be approximated by the fraction of instances in the class j among the labeled data. By substituting eq. 2 into eq. 3, we obtain:

$$p(j|x) = \frac{q_j \cdot (1 - cc_j(x))/cc_j(x)}{\sum_i q_i \cdot (1 - cc_i(x))/cc_i(x)}. \tag{4}$$

Notice that we do not have to explicitly estimate $g(x)$. Eq. 4 can be used to construct the MAP classifier:

$$\hat{c} = \arg\max_j \frac{1 - cc_j(x)}{cc_j(x)} \cdot q_j \tag{5}$$

To approximate the class-specific contrast classifier, $cc_j(x)$, we can choose any classifier that outputs a probability, such as a neural net, logistic regression, or an SVM with outputs calibrated to produce a reasonable probability.

Typically a lot more unlabeled data are available than labeled data, which causes class imbalance when training a contrast classifier. In a supervised setting, a resampling technique is often used to reduce the effect of imbalanced data. Here, we use a committee of classifiers, each of which is trained on a balanced training set sampled from each class. To compute the final output of the classifier, we implemented four different strategies.

- For each class, average the outputs of the contrast classifiers in the committee, and use the average as $cc_j(x)$ in eq. 5.

- Average only the outputs of contrast classifiers smaller than their corresponding threshold, and the fraction of the included classifiers is used as the strength of the probability output for the class.

- Use a meta classifier whose inputs are the outputs of the contrast classifiers in the committee for a class, and whose output is modeled by training it from a separate, randomly sampled data set. The output of the meta classifier is used as $cc_j(x)$.

- Classify an input as the majority class only when the outputs of the meta classifiers for the other classes are all larger than their corresponding thresholds.

Another benefit of the contrast classifier approach is that it is less affected by imbalanced data. When training the contrast classifier for each class, it uses the instances in only one class in the labeled data, and implicitly models the data distribution within that class independently of other classes. That is, given a data instance, the distribution within a class, $h_j(x)$, determines the output of the contrast classifier for the class (eq. 1), which in turn determines the posterior probability (eq. 4). Thus it will not be as highly biased toward the majority class as a classifier trained with a collection of data from imbalanced classes. Our experimental results presented in the next section confirm this benefit.

3 Experiments

We conducted experiments to answer the following questions. First, is the contrast classifier approach applicable to language processing problems, which often involve large amounts of unlabeled data? Second, does it outperform other semi-supervised learning methods on a skewed data set?

3.1 Features and data sets

The data set used consists of seven transcripts out of 75 meeting transcripts included in the ICSI meeting corpus (Janin et al., 2003). For the study, 7 meetings were segmented into spurts, defined as a chunk of speech of a speaker containing no longer than 0.5 second pause. The first 450 spurts in each of four meetings were hand-labeled as either *positive* (agreement, 9%), *negative* (disagreement, 6%), *backchannel* (23%) or *other* (62%).

To approximate $cc_j(x)$ we use a Support Vector Machine (SVM) that outputs the probability of the positive class given an instance (Lin et al., 2003). We use only word-based features similar to those used in (Hillard et al., 2003), which include the number of words in a spurt, the number of keywords associated with the *positive* and *negative* classes, and classification based on keywords. We also obtain word and class-based bigram language models for each class from the training data, and compute such language model features as the perplexity of a spurt, probability of the spurt, and the probability of the first two words in a spurt, using each language model. We also include the most likely class by the language models as features.

3.2 Results

First, we performed the same experiment as in (Hillard et al., 2003) and (Galley et al., 2004), using the contrast classifier (CC) method . Among the four meetings, the data from one meeting was set aside for testing. Table 1 compares the 3-class accuracy of the contrast classifier with previous results, merging *positive* and *backchannel* class together into one class as in the other work. When only lexical features are used (the first three entries), the SVM-based contrast classifier using meta-classifiers gives the best performance, outperforming the decision tree in (Hillard et al., 2003) and the maximum entropy model in (Galley et al., 2004). It also outperformed the SVM trained using the labeled data only. The contrast classifier is also competitive with the best case result in (Galley et al., 2004) (last entry), which adds speaker change, segment duration, and adjacency pair sequence dependency features using a dynamic Bayesian network.

Table 1: Comparison of 3-way classification accuracy on lexical (lex) vs. expanded (exp) features sets.

	Accuracy
Hillard-lex	82
Galley-lex	85.0
SVM-lex	86.3
CC-lex	86.7
Galley-exp	86.9

Table 2: Comparison of the classification performance

Method	3-way Acc	A/D confusion	A/D recovery
unsupervised	79	8	83
cc	81.4	4	82.4
cc-threshold	76.7	6	85.2
cc-meta	86.7	5	81.3
cc-meta-thres	87.1	5	82.4

In table 2, we report the performance of the four classification strategies described in section 2. For comparison, we include a result from Hillard, obtained by training a decision tree on the labels produced by their unsupervised clustering technique. Meta classifiers usually obtained higher accuracy, but averaging often achieved higher recovery of agreement/disagreement (A/D) spurts. The use of thresholds increases A/D recovery, with a decrease in accuracy. We obtained the best accuracy using both meta classifiers and thresholds together here, but we more often obtained higher accuracy using meta classifiers only.

Next, we performed experiments on the entire ICSI meeting data. Only 1,318 spurts were labeled, and 62,944 spurts were unlabeled. Again, one of the labeled meeting transcripts was set aside as a test set. We compared the SVM trained only on labeled data

Table 3: Classification performance, training on the entire ICSI data set. F is defined as $\frac{2pr}{p+r}$ where p is macro precision and r is the macro recall.

Method	Acc	F	Neg recall
SVM	85.4	72.6	21.1
self-training	80.4	65.3	5.2
cotraining	85.1	73.8	47.4
cc	83.0	75.5	68.5

with three semi-supervised methods: self-training, co-training, and the contrast classifier with a meta-classifier. The self-training iteratively trained an SVM with additional data labeled with confidence by the previously trained SVM. For the co-training, each of an SVM and a multilayer backpropagation network was trained on the labeled data and the unlabeled data classified with high confidence (99%) by one classifier were used as labeled data for further training the other classifier. We used two different classifiers, instead of two independent view of the input features as in (Goldman and Zhou, 2000). Table 3 shows that the SVM obtained high accuracy, but the F measure and the recall of the smallest class, *negative*, is quite low. The bias toward the majority class propagates through each iteration in self-training, so that only 5% of the *negative* tokens were detected after 30 iterations. We observed the same pattern in co-training; its accuracy peaked after two iterations (85.1%) and then performance degraded drastically (68% after five iterations) due in part to an increase in mislabeled data in the training set (as previously observed in (Pierce and Cardie, 2001)) and in part because the data skew is not controlled for. The contrast classifier performs better than the others in both F measure and *negative* class recall, retaining reasonably good accuracy.

4 Conclusion

In summary, our experiments on agreement/disagreement detection show that semi-supervised learning using contrast classifiers is an effective method for taking advantage of a large unlabeled data set for a problem with imbalanced classes. The contrast classifier approach outperforms co-training and self-training in detecting the infrequent classes. We also obtain good performance relative to other methods using simple lexical features and performance comparable to the best result reported.

The experiments here kept the feature set fixed, but results of (Galley et al., 2004) suggest that further gains can be achieved by augmenting the feature set. In addition, it is important to assess the impact of semi-supervised training with recognizer output, where gains from using unlabeled data may be greater than with reference transcripts as in (Hillard et al., 2003).

References

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proc. Conference on Computational Learning Theory (COLT-98)*, pages 92–100.

M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. 2004. Identifying agreement and disagreement in conversational speech: use of Bayesian networks to model dependencies. In *Proc. ACL.*

S. Goldman and Y. Zhou. 2000. Enhancing supervised learning with unlabeled data. In *Proc. the 17th ICML*, pages 327–334.

D. Hillard, M. Ostendorf, and E. Shriberg. 2003. Detection of agreement vs. disagreement in meetings: training with unlabeled data. In *Proc. HLT-NAACL.*

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The ICSI meeting corpus. In *ICASSP-03.*

T. Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proc. ICML*, pages 200–209.

H. T. Lin, C. J. Lin, and R. C. Weng. 2003. A note on platt's probabilistic outputs for support vector machines. Technical report, Dept. of Computer Science, National Taiwan University.

K. Peng, S. Vucetic, B. Han, H. Xie, and Z. Obradovic. 2003. Exploiting unlabeled data for improving accuracy of predictive data mining. In *ICDM*, pages 267–274.

D. Pierce and C. Cardie. 2001. Limitations of co-training for natural language learning from large datasets. In *Proc. EMNLP-2001).*

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. ACL*, pages 189–196.

OntoNotes: The 90% Solution

Eduard Hovy
USC/ICI
4676 Admiralty
Marina d. R., CA
hovy
@isi.edu

Mitchell Marcus
Comp & Info Science
U. of Pennsylvania
Philadelphia, PA
mitch
@cis.upenn.edu

Martha Palmer
ICS and Linguistics
U. of Colorado
Boulder, CO
martha.palmer
@colorado.edu

Lance Ramshaw
BBN Technologies
10 Moulton St.
Cambridge, MA
lance.ramshaw
@bbn.com

Ralph Weischedel
BBN Technologies
10 Moulton St.
Cambridge, MA
weischedel
@bbn.com

Abstract[*]

We describe the OntoNotes methodology and its result, a large multilingual richly-annotated corpus constructed at 90% interannotator agreement. An initial portion (300K words of English newswire and 250K words of Chinese newswire) will be made available to the community during 2007.

1 Introduction

Many natural language processing applications could benefit from a richer model of text meaning than the bag-of-words and n-gram models that currently predominate. Until now, however, no such model has been identified that can be annotated dependably and rapidly. We have developed a methodology for producing such a corpus at 90% inter-annotator agreement, and will release completed segments beginning in early 2007.

The OntoNotes project focuses on a domain independent representation of literal meaning that includes predicate structure, word sense, ontology linking, and coreference. Pilot studies have shown that these can all be annotated rapidly and with better than 90% consistency. Once a substantial and accurate training corpus is available, trained algorithms can be developed to predict these structures in new documents.

This process begins with parse (TreeBank) and propositional (PropBank) structures, which provide normalization over predicates and their arguments. Word sense ambiguities are then resolved, with each word sense also linked to the appropriate node in the Omega ontology. Coreference is also annotated, allowing the entity mentions that are propositional arguments to be resolved in context.

Annotation will cover multiple languages (English, Chinese, and Arabic) and multiple genres (newswire, broadcast news, news groups, weblogs, etc.), to create a resource that is broadly applicable.

2 Treebanking

The Penn Treebank (Marcus *et al.*, 1993) is annotated with information to make predicate-argument structure easy to decode, including function tags and markers of “empty” categories that represent displaced constituents. To expedite later stages of annotation, we have developed a parsing system (Gabbard *et al.,* 2006) that recovers both of these latter annotations, the first we know of. A first-stage parser matches the Collins (2003) parser on which it is based on the Parseval metric, while simultaneously achieving near state-of-the-art performance on recovering function tags (F-measure 89.0). A second stage, a seven stage pipeline of maximum entropy learners and voted perceptrons, achieves state-of-the-art performance (F-measure 74.7) on the recovery of empty categories by combining a linguistically-informed architecture and a rich feature set with the power of modern machine learning methods.

[*] This work was supported under the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 57–60,
New York, June 2006. ©2006 Association for Computational Linguistics

3 PropBanking

The Penn Proposition Bank, funded by ACE (DOD), focuses on the argument structure of verbs, and provides a corpus annotated with semantic roles, including participants traditionally viewed as arguments and adjuncts. The 1M word Penn Treebank II Wall Street Journal corpus has been successfully annotated with semantic argument structures for verbs and is now available via the Penn Linguistic Data Consortium as PropBank I (Palmer *et al.*, 2005). Links from the argument labels in the Frames Files to FrameNet frame elements and VerbNet thematic roles are being added. This style of annotation has also been successfully applied to other genres and languages.

4 Word Sense

Word sense ambiguity is a continuing major obstacle to accurate information extraction, summarization and machine translation. The subtle fine-grained sense distinctions in WordNet have not lent themselves to high agreement between human annotators or high automatic tagging performance. Building on results in grouping fine-grained WordNet senses into more coarse-grained senses that led to improved inter-annotator agreement (ITA) and system performance (Palmer *et al.*, 2004; Palmer *et al.*, 2006), we have developed a process for rapid sense inventory creation and annotation that includes critical links between the grouped word senses and the Omega ontology (Philpot *et al.*, 2005; see Section 5 below).

This process is based on recognizing that sense distinctions can be represented by linguists in an hierarchical structure, similar to a decision tree, that is rooted in very coarse-grained distinctions which become increasingly fine-grained until reaching WordNet senses at the leaves. Sets of senses under specific nodes of the tree are grouped together into single entries, along with the syntactic and semantic criteria for their groupings, to be presented to the annotators.

As shown in Figure 1, a 50-sentence sample of instances is annotated and immediately checked for inter-annotator agreement. ITA scores below 90% lead to a revision and clarification of the groupings by the linguist. It is only after the groupings have passed the ITA hurdle that each individual group is linked to a conceptual node in the ontology. In addition to higher accuracy, we find at least a three-fold increase in annotator productivity.

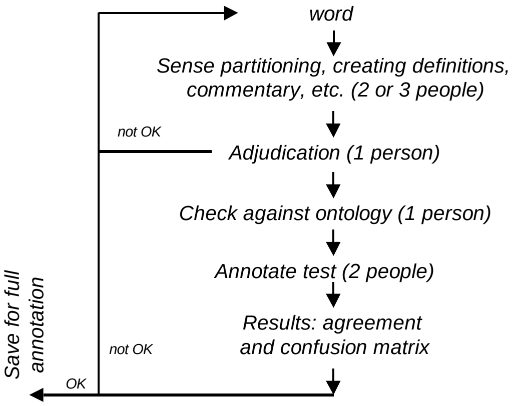

word
Sense partitioning, creating definitions, commentary, etc. (2 or 3 people)
not OK
Adjudication (1 person)
Check against ontology (1 person)
Annotate test (2 people)
Results: agreement and confusion matrix
not OK
OK
Save for full annotation

Figure 1. Annotation Procedure

As part of OntoNotes we are annotating the most frequent noun and verb senses in a 300K subset of the PropBank, and will have this data available for release in early 2007.

4.1 Verbs

Our initial goal is to annotate the 700 most frequently occurring verbs in our data, which are typically also the most polysemous; so far 300 verbs have been grouped and 150 double annotated. Subcategorization frames and semantic classes of arguments play major roles in determining the groupings, as illustrated by the grouping for the 22 WN 2.1 senses for *drive* in Figure 2. In ad-

GI: operating or traveling via a vehicle *NP (Agent) drive NP, NP drive PP*	WN1: “Can you drive a truck?”, WN2: “drive to school,”, WN3: “drive her to school,”, WN12: “this truck drives well,” WN13: “he drives a taxi,”,WN14: “The car drove around the corner,”, WN:16: “drive the turnpike to work,”
G2: force to a position or stance *NP drive NP/PP/infinitival*	WN4: “He drives me mad.,” WN6: “drive back the invaders,” WN7: “She finally drove him to change jobs,” WN8: “drive a nail,” WN15: “drive the herd,” WN22: “drive the game.”
G3: to exert energy on behalf of something *NP drive NP/infinitival*	WN5: “Her passion drives her,” WN10: “He is driving away at his thesis.”
G4: cause object to move rapidly by striking it *NP drive NP*	WN9: “drive the ball into the outfield ,” WN17 “drive a golf ball,” WN18 “drive a ball”

Figure 2. A Portion of the Grouping of WordNet Senses for "drive”

dition to improved annotator productivity and accuracy, we predict a corresponding improvement in word sense disambiguation performance. Training on this new data, Chen and Palmer (2005) report 86.3% accuracy for verbs using a smoothed maximum entropy model and rich linguistic features, which is 10% higher than their earlier, state-of-the art performance on ungrouped, fine-grained senses.

4.2 Nouns

We follow a similar procedure for the annotation of nouns. The same individual who groups WordNet verb senses also creates noun senses, starting with WordNet and other dictionaries. We aim to double-annotate the 1100 most frequent polysemous nouns in the initial corpus by the end of 2006, while maximizing overlap with the sentences containing annotated verbs.

Certain nouns carry predicate structure; these include nominalizations (whose structure obviously is derived from their verbal form) and various types of relational nouns (like *father*, *President*, and *believer*, that express relations between entities, often stated using *of*). We have identified a limited set of these whose structural relations can be semi-automatically annotated with high accuracy.

5 Ontology

In standard dictionaries, the senses for each word are simply listed. In order to allow access to additional useful information, such as subsumption, property inheritance, predicate frames from other sources, links to instances, and so on, our goal is to link the senses to an ontology. This requires decomposing the hierarchical structure into subtrees which can then be inserted at the appropriate conceptual node in the ontology.

The OntoNotes terms are represented in the 110,000-node Omega ontology (Philpot *et al.*, 2005), under continued construction and extension at ISI. Omega, which has been used for MT, summarization, and database alignment, has been assembled semi-automatically by merging a variety of sources, including Princeton's WordNet, New Mexico State University's Mikrokosmos, and a variety of Upper Models, including DOLCE (Gangemi et al., 2002), SUMO (Niles and Pease, 2001), and ISI's Upper Model, which are in the process of being reconciled. The verb frames from PropBank, FrameNet, WordNet, and Lexical Conceptual Structures (Dorr and Habash, 2001) have all been included and cross-linked.

In work planned for later this year, verb and noun sense groupings will be manually inserted into Omega, replacing the current (primarily WordNet-derived) contents. For example, of the verb groups for *drive* in the table above, G1 and G4 will be placed into the area of "controlled motion", while G2 will then sort with "attitudes".

6 Coreference

The coreference annotation in OntoNotes connects coreferring instances of specific referring expressions, meaning primarily NPs that introduce or access a discourse entity. For example, "Elco Industries, Inc.", "the Rockford, Ill. Maker of fasteners", and "it" could all corefer. (Non-specific references like "officials" in "Later, officials reported…" are not included, since coreference for them is frequently unclear.) In addition, proper premodifiers and verb phrases can be marked when coreferent with an NP, such as linking, "when the company withdrew from the bidding" to "the withdrawal of New England Electric".

Unlike the coreference task as defined in the ACE program, attributives are not generally marked. For example, the "veterinarian" NP would not be marked in "Baxter Black is a large animal veterinarian". Adjectival modifiers like "American" in "the American embassy" are also not subject to coreference.

Appositives are annotated as a special kind of coreference, so that later processing will be able to supply and interpret the implicit copula link.

All of the coreference annotation is being doubly annotated and adjudicated. In our initial English batch, the average agreement scores between each annotator and the adjudicated results were 91.8% for normal coreference and 94.2% for appositives.

7 Related and Future Work

PropBank I (Palmer *et al.*, 2005), developed at UPenn, captures predicate argument structure for verbs; NomBank provides predicate argument structure for nominalizations and other noun predicates (Meyers *et al.*, 2004). PropBank II annotation (eventuality ID's, coarse-grained sense tags, nominal coreference and selected discourse connectives) is being applied to a small (100K) parallel Chinese/English corpus (Babko-Malaya *et al.*, 2004). The OntoNotes representation extends these annotations, and allows eventual inclusion of additional shallow semantic representations for other phenomena, including temporal and spatial relations, numerical expressions, deixis, etc. One of the principal aims of OntoNotes is to enable automated semantic analysis. The best current algorithm for semantic role labeling for PropBank style annotation (Pradhan *et al.*, 2005) achieves an F-measure of 81.0 using an SVM. OntoNotes will provide a large amount of new training data for similar efforts.

Existing work in the same realm falls into two classes: the development of resources for specific phenomena or the annotation of corpora. An example of the former is Berkeley's FrameNet project (Baker *et al.*, 1998), which produces rich semantic frames, annotating a set of examples for each predicator (including verbs, nouns and adjectives), and describing the network of relations among the semantic frames. An example of the latter type is the Salsa project (Burchardt *et al.*, 2004), which produced a German lexicon based on the FrameNet semantic frames and annotated a large German newswire corpus. A second example, the Prague Dependency Treebank (Hajic *et al.*, 2001), has annotated a large Czech corpus with several levels of (tectogrammatical) representation, including parts of speech, syntax, and topic/focus information structure. Finally, the IL-Annotation project (Reeder et al., 2004) focused on the representations required to support a series of increasingly semantic phenomena across seven languages (Arabic, Hindi, English, Spanish, Korean, Japanese and French). In intent and in many details, OntoNotes is compatible with all these efforts, which may one day all participate in a larger multilingual corpus integration effort.

References

O. Babko-Malaya, M. Palmer, N. Xue, A. Joshi, and S. Kulick. 2004. Proposition Bank II: Delving Deeper, *Frontiers in Corpus Annotation, Workshop, HLT/NAACL*

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING/ACL*, pages 86-90.

J. Chen and M. Palmer. 2005. Towards Robust High Performance Word Sense Disambiguation of English Verbs Using Rich Linguistic Features. In *Proceedings of IJCNLP2005*, pp. 933-944.

B. Dorr and N. Habash. 2001. Lexical Conceptual Structure Lexicons. In Calzolari et al. ISLE-IST-1999-10647-WP2-WP3, *Survey of Major Approaches Towards Bilingual/Multilingual Lexicons.*

A. Burchardt, K. Erk, A. Frank, A. Kowalski, S. Pado, and M. Pinkal. 2006. Consistency and Coverage: Challenges for exhaustive semantic annotation. In *Proceedings of DGfS-06.*

C. Fellbaum (ed.). 1998. *WordNet: An On-line Lexical Database and Some of its Applications*. MIT Press.

R. Gabbard, M. Marcus, and S. Kulick. Fully Parsing the Penn Treebank. In *Proceedings of HLT/NAACL 2006.*

A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. 2002. Sweetening Ontologies with DOLCE. In *Proceedings of EKAW* pp. 166-181.

J. Hajic, B. Vidová-Hladká, and P. Pajas. 2001: The Prague Dependency Treebank: Annotation Structure and Support. *Proceeding of the IRCS Workshop on Linguistic Databases,* pp. 105–114.

M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics* 19: 313-330.

A. Meyers, R. Reeves, C Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank Project: An Interim Report. *Frontiers in Corpus Annotation, Workshop in conjunction with HLT/NAACL.*

I. Niles and A. Pease. 2001. Towards a Standard Upper Ontology. *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS-2001).*

M. Palmer, O. Babko-Malaya, and H. T. Dang. 2004. Different Sense Granularities for Different Applications, *2nd Workshop on Scalable Natural Language Understanding Systems, at HLT/NAACL-04,*

M. Palmer, H. Dang and C. Fellbaum. 2006. Making Fine-grained and Coarse-grained Sense Distinctions, Both Manually and Automatically, *Journal of Natural Language Engineering,* to appear.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles, *Computational Linguistics,* 31(1).

A. Philpot, E.. Hovy, and P. Pantel. 2005. The Omega Ontology. Proceedings of the ONTOLEX Workshop at IJCNLP

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, D. Jurafsky. 2005. Semantic Role Labeling Using Different Syntactic Views. *Proceedings of the ACL.*

F. Reeder, B. Dorr, D. Farwell, N. Habash, S. Helmreich, E.H. Hovy, L. Levin, T. Mitamura, K. Miller, O. Rambow, A. Siddharthan. 2004. Interlingual Annotation for MT Development. *Proceedings of AMTA.*

Investigating Cross-Language Speech Retrieval for a Spontaneous Conversational Speech Collection

Diana Inkpen, Muath Alzghool
School of Info. Technology and Eng.
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
{diana,alzghool}@site.uottawa.ca

Gareth J.F. Jones
School of Computing
Dublin City University
Dublin 9, Ireland
Gareth.Jones@computing.dcu.ie

Douglas W. Oard
College of Info. Studies/UMIACS
University of Maryland
College Park, MD 20742, USA
oard@umd.edu

Abstract

Cross-language retrieval of spontaneous speech combines the challenges of working with noisy automated transcription and language translation. The CLEF 2005 Cross-Language Speech Retrieval (CL-SR) task provides a standard test collection to investigate these challenges. We show that we can improve retrieval performance: by careful selection of the term weighting scheme; by decomposing automated transcripts into phonetic substrings to help ameliorate transcription errors; and by combining automatic transcriptions with manually-assigned metadata. We further show that topic translation with online machine translation resources yields effective CL-SR.

1 Introduction

The emergence of large collections of digitized spoken data has encouraged research in speech retrieval. Previous studies, notably those at TREC (Garafolo et al, 2000), have focused mainly on well-structured news documents. In this paper we report on work carried out for the Cross-Language Evaluation Forum (CLEF) 2005 Cross-Language Speech Retrieval (CL-SR) track (White et al, 2005). The document collection for the CL-SR task is a part of the oral testimonies collected by the USC Shoah Foundation Institute for Visual History and Education (VHI) for which some Automatic Speech Recognition (ASR) transcriptions are available (Oard et al., 2004). The data is conversional spontaneous speech lacking clear topic boundaries; it is thus a more challenging speech retrieval task than those explored previously. The CLEF data is also annotated with a range of automatic and manually generated sets of metadata. While the complete VHI dataset contains interviews in many languages, the CLEF 2005 CL-SR task focuses on English speech. Cross-language searching is evaluated by making the topic statements (from which queries are automatically formed) available in several languages. This task raises many interesting research questions; in this paper we explore alternative term weighting methods and content indexing strategies.

The remainder of this paper is structured as follows: Section 2 briefly reviews details of the CLEF 2005 CL-SR task; Section 3 describes the system we used to investigate this task; Section 4 reports our experimental results; and Section 5 gives conclusions and details for our ongoing work.

2 Task description

The CLEF-2005 CL-SR collection includes 8,104 manually-determined topically-coherent segments from 272 interviews with Holocaust survivors, witnesses and rescuers, totaling 589 hours of speech. Two ASR transcripts are available for this data, in this work we use transcripts provided by IBM Research in 2004 for which a mean word error rate of 38% was computed on held out data. Additional, metadata fields for each segment include: two sets of 20 automatically assigned thesaurus terms from different kNN classifiers (AK1 and AK2), an average of 5 manually-assigned thesaurus terms (MK), and a 3-sentence summary written by a subject matter expert. A set of 38 training topics and 25 test topics were generated in English from actual user requests. Topics were structured as Title, Description and Narrative fields, which correspond roughly to a 2-3 word Web query, what someone might first say to a librarian, and what that librarian might ultimately understand after a brief reference interview. To support CL-SR experiments the topics were re-expressed in Czech, German, French, and Spanish by native speakers in a manner reflecting

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 61–64,
New York, June 2006.

the way questions would be posed in those languages. Relevance judgments were manually generated using by augmenting an interactive search-guided procedure and purposive sampling designed to identify additional relevant segments. See (Oard et al, 2004) and (White et al, 2005) for details.

3 System Overview

Our Information Retrieval (IR) system was built with off-the-shelf components. Topics were translated from French, Spanish, and German into English using seven free online machine translation (MT) tools. Their output was merged in order to allow for variety in lexical choices. All the translations of a topic Title field were combined in a merged Title field of the translated topics; the same procedure was adopted for the Description and Narrative fields. Czech language topics were translated using InterTrans, the only web-based MT system available to us for this language pair. Retrieval was carried out using the SMART IR system (Buckley et al, 1993) applying its standard stop word list and stemming algorithm.

In system development using the training topics we tested SMART with many different term weighting schemes combining collection frequency, document frequency and length normalization for the indexed collection and topics (Salton and Buckley, 1988). In this paper we employ the notation used in SMART to describe the combined schemes: xxx.xxx. The first three characters refer to the weighting scheme used to index the document collection and the last three characters refer to the weighting scheme used to index the topic fields. For example, lpc.atc means that lpc was used for documents and atc for queries. lpc would apply log term frequency weighting (l) and probabilistic collection frequency weighting (p) with cosine normalization to the document collection (c). atc would apply augmented normalized term frequency (a), inverse document frequency weight (t) with cosine normalization (c).

One scheme in particular (mpc.ntn) proved to have much better performance than other combinations. For weighting document terms we used term frequency normalized by the maximum value (m) and probabilistic collection frequency weighting (p) with cosine normalization (c). For topics we used non-normalized term frequency (n) and inverse document frequency weighting (t) without vector normalization (n). This combination worked very well when all the fields of the query were used; it also worked well with Title plus Description, but slightly less well with the Title field alone.

4 Experimental Investigation

In this section we report results from our experimental investigation of the CLEF 2005 CL-SR task. For each set of experiments we report Mean uninterpolated Average Precision (MAP) computed using the *trec_eval* script. The topic fields used are indicated as: T for title only, TD for title + description, TDN for title + description + narrative. The first experiment shows results for different term weighting schemes; we then give cross-language retrieval results. For both sets of experiments, "documents" are represented by combining the ASR transcription with the AK1 and AK2 fields. Thus each document representation is generated completely automatically. Later experiments explore two alternative indexing strategies.

4.1 Comparison of Term Weighting Schemes

The CLEF 2005 CL-SR collection is quite small by IR standards, and it is well known that collection size matters when selecting term weighting schemes (Salton and Buckley, 1988). Moreover, the documents in this case are relatively short, averaging about 500 words (about 4 minutes of speech), and that factor may affect the optimal choice of weighting schemes as well. We therefore used the training topics to explore the space of available SMART term weighting schemes. Table 1 presents results for various weighting schemes with English topics. There are 3,600 possible combinations of weighting schemes available: 60 schemes (5 x 4 x 3) for documents and 60 for queries. We tested a total of 240 combinations. In Table 1 we present the results for 15 combinations (the best ones, plus some others to illustate the diversity of the results). mpc.ntn is still the best for the test topic set; but, as shown, a few other weighting schemes achieve similar performance. Some of the weighting schemes perform better when indexing all the topic fields (TDN), some on TD, and some on title only (T). npn.ntn was best for TD and lsn.ntn and lsn.atn are best for T. The mpc.ntn weighting scheme is used for all other experiments in this section. We are investigating the reasons for the effectiveness of this weighting scheme in our experiments.

	Weighting scheme	TDN	TD	T
		Map	Map	Map
1	Mpc.mts	0.2175	0.1651	0.1175
2	Mpc.nts	0.2175	0.1651	0.1175
3	Mpc.ntn	**0.2176**	0.1653	0.1174
4	npc.ntn	**0.2176**	0.1653	0.1174
5	Mpc.mtc	**0.2176**	0.1653	0.1174
6	Mpc.ntc	**0.2176**	0.1653	0.1174
7	Mpc.mtn	**0.2176**	0.1653	0.1174
8	Npn.ntn	0.2116	**0.1681**	0.1181
9	lsn.ntn	0.1195	0.1233	**0.1227**
10	lsn.atn	0.0919	0.1115	**0.1227**
11	asn.ntn	0.0912	0.0923	0.1062
12	snn.ntn	0.0693	0.0592	0.0729
13	sps.ntn	0.0349	0.0377	0.0383
14	nps.ntn	0.0517	0.0416	0.0474
15	Mtc.atc	0.1138	0.1151	0.1108

Table 1. MAP, 25 English test topics. Bold=best scores.

4.2 Cross-Language Experiments

Table 2 shows our results for the merged ASR, AK1 and AK2 documents with multi-system topic translations for French, German and Spanish, and single-system Czech translation. We can see that Spanish topics perform well compared to monolingual English. However, results for German and Czech are much poorer. This is perhaps not surprising for the Czech topics where only a single translation is available. For German, the quality of translation was sometimes low and some German words were retained untranslated. For French, only TD topic fields were available. In this case we can see that cross-language retrieval effectiveness is almost identical to monolingual English. Every research team participating in the CLEF 2005 CL-SR task submitted at least one TD English run, and among those our mpc.ntn system yielded the best MAP (Wilcoxon signed rank test for paired samples, $p<0.05$). However, as we show in Table 4, manual metadata can yield better retrieval effectiveness than automatic description.

Topic Language	System	Map	Fields
English	Our system	0.1653	TD
English	Our system	0.2176	TDN
Spanish	Our system	0.1863	TDN
French	Our system	0.1685	TD
German	Our system	0.1281	TDN
Czech	Our system	0.1166	TDN

Table 2. MAP, cross-language, 25 test topics

Language	Map	Fields	Description
English	0.1276	T	Phonetic
English	0.2550	TD	Phonetic
English	0.1245	T	Phonetic+Text
English	0.2590	TD	Phonetic+Text
Spanish	0.1395	T	Phonetic
Spanish	0.2653	TD	Phonetic
Spanish	0.1443	T	Phonetic+Text
Spanish	0.2669	TD	Phonetic+Text
French	0.1251	T	Phonetic
French	0.2726	TD	Phonetic
French	0.1254	T	Phonetic+Text
French	0.2833	TD	Phonetic+Text
German	0.1163	T	Phonetic
German	0.2356	TD	Phonetic
German	0.1187	T	Phonetic+Text
German	0.2324	TD	Phonetic+Text
Czech	0.0776	T	Phonetic
Czech	0.1647	TD	Phonetic
Czech	0.0805	T	Phonetic+Text
Czech	0.1695	TD	Phonetic+Text

Table 3. MAP, phonetic 4-grams, 25 test topics.

4.3 Results on Phonetic Transcriptions

In Table 3 we present results for an experiment where the text of the collection and topics, without stemming, is transformed into a phonetic transcription. Consecutive phones are then grouped into overlapping n-gram sequences (groups of n sounds, n=4 in our case) that we used for indexing. The phonetic n-grams were provided by Clarke (2005), using NIST's text-to-phone tool[1]. For example, the phonetic form for the query fragment *child survivors* is: ch_ay_l_d s_ax_r_v ax_r_v_ay r_v_ay_v v_ay_v_ax ay_v_ax_r v_ax_r_z.

The phonetic form helps compensate for the speech recognition errors. With TD queries, the results improve substantially compared with the text form of the documents and queries (9% relative). Combining phonetic and text forms (by simply indexing both phonetic n-grams and text) yields little additional improvement.

4.4 Manual summaries and keywords

Manually prepared transcripts are not available for this test collection, so we chose to use manually assigned metadata as a reference condition. To explore the effect of merging automatic and manual fields, Table 4 presents the results combining man-

[1] http://www.nist.gov/speech/tools/

ual keywords and manual summaries with ASR transcripts, AK1, and AK2. Retrieval effectiveness increased substantially for all topic languages. The MAP score improved with 25% relative when adding the manual metadata for English TDN.

Table 4 also shows comparative results between and our results and results reported by the University of Maryland at CLEF 2005 using a widely used IR system (InQuery) that has a standard term weighting algorithm optimized for large collections. For English TD, our system is 6% (relative) better and for French TD 10% (relative) better. The University of Maryland results with only automated fields are also lower than the results we report in Table 2 for the same fields.

Table 4. MAP, indexing all fields (MK, summaries, ASR transcripts, AK1 and AK2), 25 test topics.

Language	**System**	**Map**	**Fields**
English	Our system	0.4647	TDN
English	Our system	0.3689	TD
English	InQuery	0.3129	TD
English	Our system	0.2861	T
Spanish	Our system	0.3811	TDN
French	Our system	0.3496	TD
French	InQuery	0.2480	TD
French	Our system	0.3496	TD
German	Our system	0.2513	TDN
Czech	Our system	0.2338	TDN

5 Conclusions and Further Investigation

The system described in this paper obtained the best results among the seven teams that participated in the CLEF 2005 CL-SR track. We believe that this results from our use of the 38 training topics to find a term weighting scheme that is particularly suitable for this collection. Relevance judgments are typically not available for training until the second year of an IR evaluation; using a search-guided process that does not require system results to be available before judgments can be performed made it possible to accelerate that timetable in this case. Table 2 shows that performance varies markedly with the choice of weighting scheme. Indeed, some of the classic weighting schemes yielded much poorer results than the one we ultimately selected. In this paper we presented results on the test queries, but we observed similar effects on the training queries.

On combined manual and automatic data, the best MAP score we obtained for English topics is 0.4647. On automatic data, the best MAP is 0.2176. This difference could result from ASR errors or from terms added by human indexers that were not available to the ASR system to be recognized. In future work we plan to investigate methods of removing or correcting some of the speech recognition errors in the ASR transcripts using semantic coherence measures.

In ongoing further work we are exploring the relationship between properties of the collection and the weighting schemes in order to better understand the underlying reasons for the demonstrated effectiveness of the mpc.ntn weighting scheme.

The challenges of CLEF CL-SR task will continue to expand in subsequent years as new collections are introduced (e.g., Czech interviews in 2006). Because manually assigned segment boundaries are available only for English interviews, this will yield an unknown topic boundary condition that is similar to previous experiments with automatically transcribed broadcast news the Text Retrieval Conference (Garafolo et al, 2000), but with the additional caveat that topic boundaries are not known for the ground truth relevance judgments.

References

Chris Buckley, Gerard Salton, and James Allan. 1993. Automatic retrieval with locality information using SMART. In Proceedings of the First Text REtrieval Conference (TREC-1), pages 59–72.

Charles L. A. Clarke. 2005. Waterloo Experiments for the CLEF05 SDR Track, in Working Notes for the CLEF 2005 Workshop, Vienna, Austria

John S. Garofolo, Cedric G.P. Auzanne and Ellen M. Voorhees. 2000. The TREC Spoken Document Retrieval Track: A Success Story. In Proceedings of the RIAO Conference: Content-Based Multimedia Information Access, Paris, France, pages 1-20.

Douglas W. Oard, Dagobert Soergel, David Doermann, Xiaoli Huang, G. Craig Murray, Jianqiang Wang, Bhuvana Ramabhadran, Martin Franz and Samuel Gustman. 2004. Building an Information Retrieval Test Collection for Spontaneous Conversational Speech, in Proceedings of SIGIR, pages 41-48.

Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic retrieval. Information Processing and Management, 24(5):513-523.

Ryen W. White, Douglas W. Oard, Gareth J. F. Jones, Dagobert Soergel and Xiaoli Huang. 2005. Overview of the CLEF-2005 Cross-Language Speech Retrieval Track, in Working Notes for the CLEF 2005 Workshop, Vienna, Austria

Evaluating Centering for Sentence Ordering in Two New Domains

Nikiforos Karamanis
Natural Language and Information Processing Group
Computer Laboratory
University of Cambridge
Nikiforos.Karamanis@cl.cam.ac.uk

Abstract

This paper builds on recent research investigating sentence ordering in text production by evaluating the Centering-based metrics of coherence employed by Karamanis et al. (2004) using the data of Barzilay and Lapata (2005). This is the first time that Centering is evaluated empirically as a sentence ordering constraint in several domains, verifying the results reported in Karamanis et al.

1 Introduction

As most literature in text linguistics argues, a felicitous text should be *coherent* which means that the content has to be organised in a way that makes the text easy to read and comprehend. The easiest way to demonstrate this claim is by arbitrarily reordering the sentences that an understandable text consists of. This process very often gives rise to documents that do not make sense although the information content remains the same. Hence, deciding in which sequence to present a set of preselected information-bearing items is an important problem in automatic text production.

Entity coherence, which arises from the way NP referents relate subsequent sentences in the text, is an important aspect of textual felicity. *Centering Theory* (Grosz et al., 1995) has been an influential framework for modelling entity coherence in computational linguistics in the last two decades. Karamanis et al. (2004) were the first to evaluate Centering-based metrics of coherence for ordering clauses in a subset of the GNOME corpus (Poesio et al., 2004) consisting of 20 artefact descriptions. They introduced a novel experimental methodology that treats the observed ordering of clauses in a text as the gold standard, which is scored by each metric. Then, the metric is penalised proportionally to the amount of alternative orderings of the same material that score equally to or better than the gold standard.

This methodology is very similar to the way Barzilay and Lapata (2005) evaluate automatically another model of coherence called the entity grid using a larger collection of 200 articles from the North American News Corpus (NEWS) and 200 accident narratives from the National Transportation Safety Board database (ACCS). The same data and similar methods were used by Barzilay and Lee (2004) to compare their probabilistic approach for ordering sentences with that of Lapata (2003).

This paper discusses how the Centering-based metrics of coherence employed by Karamanis et al. can be evaluated on the data prepared by Barzilay and Lapata. This is the first time that Centering is evaluated empirically as a sentence ordering constraint in more than one domain, verifying the results reported in Karamanis et al.

The paper also contributes by emphasising the following methodological point: To conduct our experiments, we need to produce several alternative orderings of sentences and compare them with the gold standard. As the number of possible orderings grows factorially, enumerating them exhaustively (as Barzilay and Lee do) becomes impractical. In this paper, we make use of the methods of Karamanis (2003) which allow us to explore a

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 65–68,
New York, June 2006. ©2006 Association for Computational Linguistics

Table 1A	NP referents						
Sentences	department	trial	microsoft	...	products	brands	...
(a)	S	O	S	...	–	–	...
(b)	–	–	O	...	S	O	...

Table 1B	CF list:			CHEAPNESS
Sentences	{CP, next two referents}	CB	Transition	CB$_n$=CP$_{n-1}$
(a)	{department, microsoft, trial, ...}	n.a.	n.a.	n.a.
(b)	{products, microsoft, brands, ...}	microsoft	RETAIN	$*$

Table 1: (A) Fragment of the entity grid for example (1); (B) CP (i.e. first member of the CF list), next two referents, CB, transition and violations of CHEAPNESS (denoted with a $*$) for the same example.

sufficient number of alternative orderings and return more reliable results than Barzilay and Lapata, who used a sample of just 20 randomly produced orderings (often out of several millions).

2 Materials and methods

2.1 Centering data structures

Example (1) presents the first two sentences of a text in NEWS (Barzilay and Lapata, Table 2):

(1) (a) [The Justice Department]$_S$ is conducting [an antitrust trial]$_O$ against [Microsoft Corp.]$_X$ with [evidence]$_X$ that [the company]$_S$ is increasingly attempting to crush [competitors]$_O$. (b) [Microsoft]$_O$ is accused of trying to forcefully buy into [markets]$_X$ where [its own products]$_S$ are not competitive enough to unseat [established brands]$_O$. (...)

Barzilay and Lapata automatically annotated their corpora for the grammatical role of the NPs in each sentence (denoted in the example by the subscripts S, O and X for subject, object and other respectively)[1] as well as their coreferential relations. This information is used as the basis for the computation of the entity grid: a two-dimensional array that captures the distribution of NP referents across sentences in the text using the aforementioned symbols for their grammatical role and "–" for a referent that does not occur in a sentence. Table 1A illustrates a fragment of the grid for the sentences in example (1).[2]

Our data transformation script computes the basic structure of Centering (known as CF list) for each row of the grid using the referents with the symbols S, O and X (Table 1B). The members of the CF list are ranked according to their grammatical role (Brennan et al., 1987) and their position in the grid.[3] The derived sequence of CF lists can then be used to compute other important Centering concepts:

- The CB, i.e. the referent that links the current CF list with the previous one such as microsoft in (b).
- Transitions (Brennan et al., 1987) and NOCBs, that is, cases in which two subsequent CF lists do not have any referent in common.
- Violations of CHEAPNESS (Strube and Hahn, 1999), COHERENCE and SALIENCE (Kibble and Power, 2000).

2.2 Metrics of coherence

Karamanis (2003) assumes a system which receives an unordered set of CF lists as its input and uses a metric to output the highest scoring ordering. He discusses how Centering can be used to define many different metrics of coherence which might be useful for this task. In our experiments we made use of the four metrics employed in Karamanis et al. (2004):

- The baseline metric M.NOCB which simply prefers the ordering with the fewest NOCBs.
- M.CHEAP which selects the ordering with the fewest violations of CHEAPNESS.
- M.KP, introduced by Kibble and Power, which sums up the NOCBs as well as the violations of CHEAPNESS, COHERENCE and SALIENCE, preferring the ordering with the lowest total cost.
- M.BFP which employs the transition preferences of Brennan et al.

[1] Subjects in passive constructions such as "Microsoft" in (1b) are marked with O.

[2] If a referent such as microsoft is attested by several NPs, e.g. "Microsoft Corp." and "the company" in (1a), the role with the highest priority (in this case S) is used.

[3] The referent department appears in an earlier grid column than microsoft because "the Justice Department" is mentioned before "Microsoft Corp." in the text. Since grid position corresponds to order of mention, the former can be used to resolve ties between referents with the same grammatical role in the CF list similarly to the use of the latter e.g. by Strube and Hahn.

NEWS	M.NOCB			p
corpus	lower	greater	ties	
M.CHEAP	155	44	1	<0.000
M.KP	131	68	1	<0.000
M.BFP	121	71	8	<0.000
N of texts	200			

Table 2: Comparing M.NOCB with M.CHEAP, M.KP and M.BFP in the NEWS corpus.

2.3 Experimental methodology

As already mentioned, previous work assumes that the gold standard ordering (GSO) observed in a text is more coherent than any other ordering of the sentences (or the corresponding CF lists) it consists of. If a metric takes a randomly produced ordering to be more coherent than the GSO, it has to be penalised.

Karamanis et al. (2004) introduce a measure called the *classification rate* which estimates this penalty as the weighted sum of the percentage of alternative orderings that score equally to or better than the GSO.[4] When comparing several metrics with each other, the one with the lowest classification rate is the most appropriate for sentence ordering.

Karamanis (2003) argues that computing the classification rate using a random sample of one million orderings provides reliable results for the entire population of orderings. In our experiments, we used a random sample of that size for GSOs which consisted of more than 10 sentences. This allows us to explore a sufficient portion of possible orderings (without having to exhaustively enumerate every ordering as Barzilay and Lee do). Arguably, our experiments also return more reliable results than those of Barzilay and Lapata who used a sample of just a few randomly produced orderings.

Since the Centering-based metrics can be directly deployed on unseen texts without any training, we treated all texts in NEWS and ACCS as testing data.[5]

ACCS	M.NOCB			p
corpus	lower	greater	ties	
M.CHEAP	183	17	0	<0.000
M.KP	167	33	0	<0.000
M.BFP	100	100	0	1.000
N of texts	200			

Table 3: Comparing M.NOCB with M.CHEAP, M.KP and M.BFP in the ACCS corpus.

3 Results

The experimental results of the comparisons of the metrics from section 2.2 are reported in Table 2 for the NEWS corpus and in Table 3 for ACCS. Following Karamanis et al., the tables compare the baseline metric M.NOCB with each of M.CHEAP, M.KP and M.BFP. The exact number of GSOs for which the classification rate of M.NOCB is lower than its competitor for each comparison is reported in the second column of the Table. For example, M.NOCB has a lower classification rate than M.CHEAP for 155 (out of 200) GSOs from NEWS. M.CHEAP achieves a lower classification rate for just 44 GSOs, while there is a single tie in which the classification rate of the two metrics is the same. The p value returned by the two-tailed sign test for the difference in the number of GSOs, rounded to the third decimal place, is reported in the fifth column of Table 2.[6]

Overall, the Table shows that M.NOCB does significantly better in NEWS than the other three metrics which employ additional Centering concepts. Similarly, M.CHEAP and M.KP are overwhelmingly beaten by the baseline in ACCS. Also note that since M.BFP fails to significantly overtake M.NOCB in ACCS, the baseline can be considered the most promising solution in that case too by applying Occam's razor.

Table 4 shows the results of the evaluation of the metrics in GNOME from Karamanis et al. These results are strikingly similar to ours despite the much smaller size of their sample. Hence, M.NOCB is the most suitable among the investigated metrics for ordering the CF lists in both NEWS and ACCS in addition to GNOME.

[4] The classification rate is computed according to the formula Better(M,GSO) + Equal(M,GSO)/2. Better(M,GSO) stands for the percentage of orderings that score better than the GSO according to a metric M, whilst Equal(M,GSO) is the percentage of orderings that score equal to the GSO.

[5] By contrast, Barzilay and Lapata used 100 texts in each domain to train their probabilistic model and 100 to test it. Note that although they experiment with quite large corpora their reported results are not verified by statistical tests.

[6] The sign test was chosen by Karamanis et al. to test significance because it does not carry specific assumptions about population distributions and variance.

GNOME	M.NOCB			p
corpus	lower	greater	ties	
M.CHEAP	18	2	0	<0.000
M.KP	16	2	2	0.002
M.BFP	12	3	5	0.036
N of texts	20			

Table 4: Comparing M.NOCB with M.CHEAP, M.KP and M.BFP in the GNOME corpus.

4 Discussion

Our experiments have shown that the baseline M.NOCB performs better than its competitors. This in turn indicates that simply avoiding NOCB transitions is more relevant to sentence ordering than the additional Centering concepts employed by the other metrics.

But how likely is M.NOCB to come up with the GSO if it is actually used to guide an algorithm which orders the CF lists in our corpora? The *average classification rate* of M.NOCB is an estimate of exactly this variable.

The average classification rate for M.NOCB is 30.90% in NEWS and 15.51% in ACCS. The previously reported value for GNOME is 19.95%.[7] This means that on average M.NOCB takes approximately 1 out of 3 alternative orderings in NEWS and 1 out of 6 in ACCS to be more coherent that the GSO. As already observed by Karamanis et al., there results suggest that M.NOCB cannot be put in practical use.

However, the fact that M.NOCB is shown to overtake its Centering-based competitors across several corpora means that it is a simple, yet robust, baseline against which other similar metrics can be tested. For instance, Barzilay and Lapata report a ranking accuracy of around 90% for their best grid-based sentence ordering method, which we take to correspond to a classification rate of approximately 10% (assuming that there do not exist any equally scoring alternative orderings). This amounts to an improvement over M.NOCB of almost 5% in ACCS and 20% in NEWS.

Given the deficiencies of the evaluation in Barzilay and Lapata, this comparison can only be provisional. In our future work, we intend to directly evaluate their method using a substantially large number of alternative orderings and M.NOCB as the baseline. We will also try to supplement M.NOCB with other features of coherence to improve its performance.

Acknowledgments

Many thanks to Regina Barzilay and Mirella Lapata for their data, to Le An Ha for the data transformation script and to Chris Mellish, Massimo Poesio and three anonymous reviewers for comments. Support from the Rapid Item Generation project (Wolverhampton University) and the BBSRC-funded Flyslip grant (No 16291) is also acknowledged.

References

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of ACL 2005*, pages 141–148.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models with applications to generation and summarization. In *Proceedings of HLT-NAACL 2004*, pages 113–120.

Susan E. Brennan, Marilyn A. Friedman [Walker], and Carl J. Pollard. 1987. A centering approach to pronouns. In *Proceedings of ACL 1987*, pages 155–162, Stanford, California.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. 2004. Evaluating centering-based metrics of coherence using a reliably annotated corpus. In *Proceedings of ACL 2004*, pages 391–398, Barcelona, Spain.

Nikiforos Karamanis. 2003. *Entity Coherence for Descriptive Text Structuring*. Ph.D. thesis, Division of Informatics, University of Edinburgh.

Rodger Kibble and Richard Power. 2000. An integrated framework for text planning and pronominalisation. In *Proceedings of INLG 2000*, pages 77–84, Israel.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of ACL 2003*, pages 545–552, Saporo, Japan, July.

Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman. 2004. Centering: a parametric theory and its instantiations. *Computational Linguistics*, 30(3):309–363.

Michael Strube and Udo Hahn. 1999. Functional centering: Grounding referential coherence in information structure. *Computational Linguistics*, 25(3):309–344.

[7] The variability is presumably due to the different characteristics of each corpus (which do not prevent M.NOCB from always beating its competitors).

MMR-based Active Machine Learning for Bio Named Entity Recognition

Seokhwan Kim[1] **Yu Song**[2] **Kyungduk Kim**[1] **Jeong-Won Cha**[3] **Gary Geunbae Lee**[1]
[1] Dept. of Computer Science and Engineering, POSTECH, Pohang, Korea
[2] AIA Information Technology Co., Ltd. Beijing, China
[3] Dept. of Computer Science, Changwon National University, Changwon, Korea
megaup@postech.ac.kr, Song-Y.Song@AIG.com, getta@postech.ac.kr
jcha@changwon.ac.kr, gblee@postech.ac.kr

Abstract

This paper presents a new active learning paradigm which considers not only the uncertainty of the classifier but also the diversity of the corpus. The two measures for uncertainty and diversity were combined using the MMR (Maximal Marginal Relevance) method to give the sampling scores in our active learning strategy. We incorporated MMR-based active machine-learning idea into the biomedical named-entity recognition system. Our experimental results indicated that our strategies for active-learning based sample selection could significantly reduce the human effort.

1 Introduction

Named-entity recognition is one of the most elementary and core problems in biomedical text mining. To achieve good recognition performance, we use a supervised machine-learning based approach which is a standard in the named-entity recognition task. The obstacle of supervised machine-learning methods is the lack of the annotated training data which is essential for achieving good performance. Building a training corpus manually is time consuming, labor intensive, and expensive. Creating training corpora for the biomedical domain is particularly expensive as it requires domain specific expert knowledge.

One way to solve this problem is through active learning method to select the most informative samples for training. Active selection of the training examples can significantly reduce the necessary number of labeled training examples without degrading the performance.

Existing work for active learning explores two approaches: certainty or uncertainty-based methods (Lewis and Gale 1994; Scheffer and Wrobel 2001; Thompson *et al.* 1999) and committee-based methods (Cohn *et al.* 1994; Dagan and Engelson 1995; Freund *et al.* 1997; Liere and Tadepalli 1997). Uncertainty-based systems begin with an initial classifier and the systems assign some uncertainty scores to the un-annotated examples. The *k* examples with the highest scores will be annotated by human experts and the classifier will be retrained. In the committee-based systems, diverse committees of classifiers were generated. Each committee member will examine the un-annotated examples. The degree of disagreement among the committee members will be evaluated and the examples with the highest disagreement will be selected for manual annotation.

Our efforts are different from the previous active learning approaches and are devoted to two aspects: we propose an entropy-based measure to quantify the uncertainty that the current classifier holds. The most uncertain samples are selected for human annotation. However, we also assume that the selected training samples should give the different aspects of learning features to the classification system. So, we try to catch the most representative sentences in each sampling. The divergence measures of the two sentences are for the novelty of the features and their representative levels, and are described by the minimum similarity among the examples. The two measures for uncertainty and diversity will be combined using the MMR (Maximal Marginal Relevance) method (Carbonell and Goldstein 1998) to give the sampling scores in our active learning strategy.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 69–72,
New York, June 2006.

We incorporate MMR-based active machine-learning idea into the POSBIOTM/NER (Song *et al.* 2005) system which is a trainable biomedical named-entity recognition system using the Conditional Random Fields (Lafferty *et al.* 2001) machine learning technique to automatically identify different sets of biological entities in the text.

2 MMR-based Active Learning for Biomedical Named-entity Recognition

2.1 Active Learning

We integrate active learning methods into the POSBIOTM/NER (Song *et al.* 2005) system by the following procedure: Given an active learning scoring strategy *S* and a threshold value *th*, at each iteration *t*, the learner uses training corpus T_{M_t} to train the NER module M_t. Each time a user wants to annotate a set of un-labeled sentences *U*, the system first tags the sentences using the current NER module M_t. At the same time, each tagged sentence is assigned with a score according to our scoring strategy *S*. Sentences will be marked if its score is larger than the threshold value *th*. The tag result is presented to the user, and those marked ones are rectified by the user and added to the training corpus. Once the training data accumulates to a certain amount, the NER module M_t will be retrained.

2.2 Uncertainty-based Sample Selection

We evaluate the uncertainty degree that the current NER module holds for a given sentence in terms of the entropy of the sentence. Given an input sequence **o**, the state sequence set *S* is a finite set. And $p_{\wedge}(\mathbf{s}\,|\,\mathbf{o}),\ \mathbf{s}\in\mathbf{S}$ is the probability distribution over *S*. By using the equation for CRF (Lafferty *et al.* 2001) module, we can calculate the probability of any possible state sequence s given an input sequence **o**. Then the entropy of $p_{\wedge}(\mathbf{s}\,|\,\mathbf{o})$ is defined to be:

$$H = -\sum_{\mathbf{s}} P_{\wedge}(\mathbf{s}\,|\,\mathbf{o})\log_2[P_{\wedge}(\mathbf{s}\,|\,\mathbf{o})]$$

The number of possible state sequences grows exponentially as the sentence length increases. In order to measure the uncertainty by entropy, it is inconvenient and unnecessary to compute the probability of all the possible state sequences. Instead we implement N-best Viterbi search to find the *N* state sequences with the highest probabilities. The entropy *H*(*N*) is defined as the entropy of the distribution of the N-best state sequences:

$$H(N) = -\sum_{i=1}^{N} \frac{P_{\wedge}(\mathbf{s}_i\,|\,\mathbf{o})}{\sum_{i=1}^{N} P_{\wedge}(\mathbf{s}_i\,|\,\mathbf{o})} \log_2\left[\frac{P_{\wedge}(\mathbf{s}_i\,|\,\mathbf{o})}{\sum_{i=1}^{N} P_{\wedge}(\mathbf{s}_i\,|\,\mathbf{o})}\right]. \quad (1)$$

The range of the entropy *H*(*N*) is [0, $-\log_2 \frac{1}{N}$] which varies according to different *N*. We could use the equation (2) to normalize the *H*(*N*) to [0, 1].

$$H(N)' = \frac{H(N)}{-\log_2 \frac{1}{N}}. \quad (2)$$

2.3 Diversity-based Sample Selection

We measure the sentence structure similarity to represent the diversity and catch the most representative ones in order to give more diverse features to the machine learning-based classification systems.

We propose a three-level hierarchy to represent the structure of a sentence. The first level is NP chunk, the second level is Part-Of-Speech tag, and the third level is the word itself. Each word is represented using this hierarchy structure. For example in the sentence "I am a boy", the word "boy" is represented as $\vec{w}$ =[NP, NN, boy]. The similarity score of two words is defined as:

$$sim(\vec{w}_1 \cdot \vec{w}_2) = \frac{2 * Depth(\vec{w}_1, \vec{w}_2)}{Depth(\vec{w}_1) + Depth(\vec{w}_2)}$$

Where $Depth(\vec{w}_1, \vec{w}_2)$ is defined from the top level as the number of levels that the two words are in common. Under our three-level hierarchy scheme above, each word representation has depth of 3.

The structure of a sentence S is represented as the word representation vectors $[\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_N]$. We measure the similarity of two sentences by the standard cosine-similarity measure. The similarity score of two sentences is defined as:

$$similarity(\vec{S}_1, \vec{S}_2) = \frac{\vec{S}_1 \cdot \vec{S}_2}{\sqrt{\vec{S}_1 \cdot \vec{S}_1}\sqrt{\vec{S}_2 \cdot \vec{S}_2}},$$

$$\vec{S}_1 \cdot \vec{S}_2 = \sum_i \sum_j sim(\vec{w}_{1i} \cdot \vec{w}_{2j}).$$

2.4 MMR Combination for Sample Selection

We would like to score the sample sentences with respect to both the uncertainty and the diversity. The following MMR (Maximal Marginal Relevance) (Carbonell and Goldstein 1998) formula is used to calculate the active learning score:

$$score(s_i) \overset{def}{=} \lambda * \text{Uncertainty}(s_i, M) - (1-\lambda) * \max_{s_j \in T_M} \text{Similarity}(s_i, s_j) \quad (3)$$

where s_i is the sentence to be selected, Uncertainty is the entropy of s_i given current NER module *M*, and Similarity indicates the divergence degree between the s_i and the sentence s_j in the training corpus T_M of *M*. The combination rule could be interpreted as assigning a higher score to a sentence of which the NER module is uncertain and whose configuration differs from the sentences in the existing training corpus. The value of parameter λ coordinates those two different aspects of the desirable sample sentences.

After initializing a NER module M and an appropriate value of the parameter λ , we can assign each candidate sentence a score under the control of the uncertainty and the diversity.

3 Experiment and Discussion

3.1 Experiment Setup

We conducted our active learning experiments using pool-based sample selection (Lewis and Gale 1994). The pool-based sample selection, in which the learner chooses the best instances for labeling from a given pool of unlabelled examples, is the most practical approach for problems in which unlabelled data is relatively easily available.

For our empirical evaluation of the active learning methods, we used the training and test data released by JNLPBA (Kim *et al.* 2004). The training corpus contains 2000 MEDLINE abstracts, and the test data contains 404 abstracts from the GENIA corpus. 100 abstracts were used to train our initial NER module. The remaining training data were taken as the pool. Each time, we chose *k* examples from the given pool to train the new NER module and the number *k* varied from 1000 to 17000 with a step size 1000.

We test 4 different active learning methods: Random selection, Entropy-based uncertainty selection, Entropy combined with Diversity, and Normalized Entropy (equation (2)) combined with Diversity. When we compute the active learning score using the entropy based method and the combining methods we set the values of parameter N (from equation (1)) to 3 and λ (from equation (3)) to 0.8 empirically.

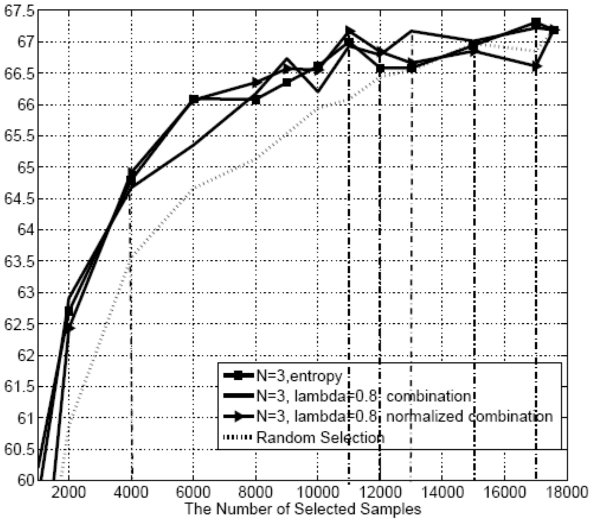


Fig1. Comparison of active learning strategies with the random selection

3.2 Results and Analyses

The initial NER module gets an F-score of 52.54, while the F-score performance of the NER module using the whole training data set is 67.19. We plotted the learning curves for the different sample selection strategies. The interval in the x-axis between the curves shows the number of examples selected and the interval in the y-axis shows the performance improved.

We compared the entropy, entropy combined with sentence diversity, normalized entropy combined with sentence diversity and random selection.

The curves in Figure 1 show the relative performance. The F-score increases along with the number of selected examples and receives the best performance when all the examples in the pool are selected. The results suggest that all three kinds of active learning strategies consistently outperform the random selection.

The entropy-based example selection has improved performance compared with the random selection. The entropy (N=3) curve approaches to the random selection around 13000 sentences selected, which is reasonable since all the methods choose the examples from the same given pool. As the number of selected sentences approaches the pool size, the performance difference among the different methods gets small. The best performance of the entropy strategy is 67.31 when 17000 examples are selected.

Comparing with the entropy curve, the combined strategy curve shows an interesting characteristic. Up to 4000 sentences, the entropy strategy and the combined strategy perform similarly. After the 11000 sentence point, the combined strategy surpasses the entropy strategy. It accords with our belief that the diversity increases the classifier's performance when the large amount of samples is selected. The normalized combined strategy differs from the combined strategy. It exceeds the other strategies from the beginning and maintains the best performance up until 12000 sentence point.

The entropy strategy reaches 67.00 in F-score when 11000 sentences are selected. The combined strategy receives 67.17 in F-score while 13000 sentences are selected, while the end performance is 67.19 using the whole training data. The combined strategy reduces 24.64 % of training examples compared with the random selection. The normalized combined strategy achieves 67.17 in F-score when 11000 sentences are selected, so 35.43% of the training examples do not need to be labeled to achieve almost the same performance as the end performance. The normalized combined strategy's performance becomes similar to the random selection strategy at around 13000 sentences, and after 14000 sentences the normalized combined strategy behaves the worst.

4 Conclusion

We incorporate active learning into the biomedical named-entity recognition system to enhance the system's performance with only small amount of training data. We presented the entropy-based uncertainty sample selection and combined selection strategies using the corpus diversity. Experiments indicate that our strategies for active-learning based sample selection could significantly reduce the human effort.

Acknowledgement

This research was supported as a Brain Neuroinformatics Research Program sponsored by Ministry of Commerce, Industry and Energy.

References

Carbonell J., & Goldstein J. (1998). The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 335-336.

Cohn, D. A., Atlas, L., & Ladner, R. E. (1994). Improving generalization with active learning, *Machine Learning*, 15(2), 201-221.

Dagan, I., & Engelson S. (1995). Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 150-157, San Francisco, CA, Morgan Kaufman.

Freund Y., Seung H.S., Shamir E., & Tishby N. (1997). Selective sampling using the query by committee algorithm, *Machine Learning*, 28, 133-168.

Kim JD., Ohta T., Tsuruoka Y., & Tateisi Y. (2004). Introduction to the Bio-Entity Recognition Task at JNLPBA, *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Application (JNLPBA).*

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the 18th International Conf. on Machine Learning*, pages 282-289, Williamstown, MA, Morgan Kaufmann.

Lewis D., & Gale W. (1994). A Sequential Algorithm for Training Text Classifiers, In: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. pp. 3-12, Springer-Verlag.

Liere, R., & Tadepalli, P. (1997). Active learning with committees for text categorization, In *proceedings of the Fourteenth National Conference on Artificial Intelligence*, pp. 591-596 Providence, RI.

Scheffer T., & Wrobel S. (2001). Active learning of partially hidden markov models. In *Proceedings of the ECML/PKDD Workshop on Instance Selection*.

Song Y., Kim E., Lee G.G., & Yi B-k. (2005). POSBIOTM-NER: a trainable biomedical named-entity recognition system. *Bioinformatics*, 21 (11): 2794-2796.

Thompson C.A., Califf M.E., & Mooney R.J. (1999). Active Learning for Natural Language Parsing and Information Extraction, In *Proceedings of the Sixteenth International Machine Learning Conference*, pp.406-414, Bled, Slovenia.

Early Deletion of Fillers In Processing Conversational Speech

Matthew Lease and Mark Johnson
Brown Laboratory for Linguistic Information Processing (BLLIP)
Brown University
Providence, RI 02912
{mlease,mj}@cs.brown.edu

Abstract

This paper evaluates the benefit of deleting fillers (e.g. *you know, like*) early in parsing conversational speech. Readability studies have shown that disfluencies (fillers and speech repairs) may be deleted from transcripts without compromising meaning (Jones et al., 2003), and deleting repairs prior to parsing has been shown to improve its accuracy (Charniak and Johnson, 2001). We explore whether this strategy of early deletion is also beneficial with regard to fillers. Reported experiments measure the effect of early deletion under in-domain and out-of-domain parser training conditions using a state-of-the-art parser (Charniak, 2000). While early deletion is found to yield only modest benefit for in-domain parsing, significant improvement is achieved for out-of-domain adaptation. This suggests a potentially broader role for disfluency modeling in adapting text-based tools for processing conversational speech.

1 Introduction

This paper evaluates the benefit of deleting fillers early in parsing conversational speech. We follow LDC (2004) conventions in using the term *filler* to encompass a broad set of vocalized space-fillers that can introduce syntactic (and semantic) ambiguity. For example, in the questions

```
Did you know I do that?
Is it like that one?
```

colloquial use of fillers, indicated below through use of commas, can yield alternative readings

```
Did, you know, I do that?
Is it, like, that one?
```

Readings of the first example differ in querying listener knowledge versus speaker action, while readings of the second differ in querying similarity versus exact match. Though an engaged listener rarely has difficulty distinguishing between such alternatives, studies show that deleting disfluencies from transcripts improves readability with no reduction in reading comprehension (Jones et al., 2003).

The fact that disfluencies can be completely removed without compromising meaning is important. Earlier work had already made this claim regarding speech repairs[1] and argued that there was consequently little value in syntactically analyzing repairs or evaluating our ability to do so (Charniak and Johnson, 2001). Moreover, this work showed that collateral damage to parse accuracy caused by repairs could be averted by deleting them prior to parsing, and this finding has been confirmed in subsequent studies (Kahn et al., 2005; Harper et al., 2005). But whereas speech repairs have received significant attention in the parsing literature, fillers have been relatively neglected. While one study has shown that the presence of interjection and parenthetical constituents in conversational speech reduces parse accuracy (Engel et al., 2002), these constituent types are defined to cover both fluent and disfluent speech phenomena (Taylor, 1996), leaving the impact of fillers alone unclear.

In our study, disfluency annotations (Taylor, 1995) are leveraged to identify fillers precisely, and these annotations are merged with treebank syntax. Extending the arguments of Charniak and Johnson with regard to repairs (2001), we argue there is little value in recovering the syntactic structure

[1] See (Core and Schubert, 1999) for a prototypical counter-example that rarely occurs in practice.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 73–76,
New York, June 2006. ©2006 Association for Computational Linguistics

of fillers, and we relax evaluation metrics accordingly (§3.2). Experiments performed (§3.3) use a state-of-the-art parser (Charniak, 2000) to study the impact of early filler deletion under in-domain and out-of-domain (i.e. adaptation) training conditions. In terms of adaptation, there is tremendous potential in applying textual tools and training data to processing transcribed speech (e.g. machine translation, information extraction, etc.), and *bleaching* speech data to more closely resemble text has been shown to improve accuracy with some text-based processing tasks (Rosenfeld et al., 1995). For our study, a state-of-the-art filler detector (Johnson et al., 2004) is employed to delete fillers prior to parsing. Results show parse accuracy improves significantly, suggesting disfluency filtering may have a broad role in enabling text-based processing of speech data.

2 Disfluency in Brief

In this section we give a brief introduction to disfluency, providing an excerpt from Switchboard (Graff and Bird, 2000) that demonstrates typical production of repairs and fillers in conversational speech.

We follow previous work (Shriberg, 1994) in describing a repair in terms of three parts: the *reparandum* (the material repaired), the corrected *alteration*, and between these an optional *interregnum* (or editing term) consisting of one or more fillers. Our notion of fillers encompasses filled pauses (e.g. `uh, um, ah`) as well as other vocalized space-fillers annotated by LDC (Taylor, 1995), such as `you know, i mean, like, so, well`, etc. Annotations shown here are typeset with the following conventions: **fillers** are bold, [reparanda] are square-bracketed, and <u>alterations</u> are underlined.

> S1: **Uh** first **um** i need to know **uh** how do you feel [about] **uh** <u>about</u> sending **uh** an elderly **uh** family member to a nursing home
>
> S2: **Well** of course [it's] **you know** <u>it's</u> one of the last few things in the world you'd ever want to do **you know** unless it's just **you know** really **you know uh** [for their] **uh you know** <u>for their</u> own good

Though disfluencies rarely complicate understanding for an engaged listener, deleting them from transcripts improves readability with no reduction in reading comprehension (Jones et al., 2003). For automated analysis of speech data, this means we may freely explore processing alternatives which delete disfluencies without compromising meaning.

3 Experiments

This section reports parsing experiments studying the effect of early deletion under in-domain and out-of-domain parser training conditions using the August 2005 release of the Charniak parser (2000). We describe data and evaluation metrics used, then proceed to describe the experiments.

3.1 Data

Conversational speech data was drawn from the Switchboard corpus (Graff and Bird, 2000), which annotates disfluency (Taylor, 1995) as well as syntax. Our division of the corpus follows that used in (Charniak and Johnson, 2001). Speech recognizer (ASR) output is approximated by removing punctuation, partial words, and capitalization, but we do use reference words, representing an upperbound condition of perfect ASR. Likewise, annotated sentence boundaries are taken to represent oracle boundary detection. Because fillers are annotated only in disfluency markup, we perform an automatic tree transform to merge these two levels of annotation: each span of contiguous filler words were pruned from their corresponding tree and then reinserted at the same position under a flat FILLER constituent, attached as highly as possible. Transforms were achieved using TSurgeon[2] and Lingua::Treebank[3].

For our out-of-domain training condition, the parser was trained on sections 2-21 of the Wall Street Journal (WSJ) corpus (Marcus et al., 1993). Punctuation and capitalization were removed to bleach our our textual training data to more closely resemble speech (Rosenfeld et al., 1995). We also tried automatically changing numbers, symbols, and abbreviations in the training text to match how they would be read (Roark, 2002), but this did not improve accuracy and so is not discussed further.

3.2 Evaluation Metrics

As discussed earlier (§1), Charniak and Johnson (2001) have argued that speech repairs do not

[2] http://nlp.stanford.edu/software/tsurgeon.shtml

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

contribute to meaning and so there is little value in syntactically analyzing repairs or evaluating our ability to do so. Consequently, they *relaxed* standard PARSEVAL (Black et al., 1991) to treat EDITED constituents like punctuation: adjacent EDITED constituents are merged, and the internal structure and attachment of EDITED constituents is not evaluated. We propose generalizing this approach to disfluency at large, i.e. fillers as well as repairs. Note that the details of appropriate evaluation metrics for parsed speech data is orthogonal to the parsing methods proposed here: however parsing is performed, we should avoid wasting metric attention evaluating syntax of words that do not contribute toward meaning and instead evaluate only how well such words can be identified.

Relaxed metric treatment of disfluency was achieved via simple parameterization of the SParseval tool (Harper et al., 2005). SParseval also has the added benefit of calculating a dependency-based evaluation alongside PARSEVAL's bracket-based measure. The dependency metric performs syntactic head-matching for each word using a set of given head percolation rules (derived from Charniak's parser (2000)), and its relaxed formulation ignores terminals spanned by FILLER and EDITED constituents. We found this metric offered additional insights in analyzing some of our results.

3.3 Results

In the first set of experiments, we train the parser on Switchboard and contrast early deletion of disfluencies (identified by an oracle) versus parsing in the more usual fashion. Our method for early deletion generalizes the approach used with repairs in (Charniak and Johnson, 2001): contiguous filler and edit words are deleted from the input strings, the strings are parsed, and the removed words are reinserted into the output trees under the appropriate flat constituent, FILLER or EDITED.

Results in Table 1 give F-scores for PARSEVAL and dependency-based parse accuracy (§3.2), as well as per-word edit and filler detection accuracy (i.e. how well the parser does in identifying which terminals should be spanned by EDITED and FILLER constituents when early deletion is not performed). We see that the parser correctly identifies filler words with 93.1% f-score, and that early deletion of fillers (via oracle knowledge) yields only a modest improvement in parsing accuracy (87.8% to 88.9% bracket-based, 87.9% to 88.5% dependency-based). We conclude from this that for in-domain training, early deletion of fillers has limited potential to improve parsing accuracy relative to what has been seen with repairs. It is still worth noting, however, that the parser does perform better when fillers are absent, consistent with Engel et al.'s findings (2002). While fillers have been reported to often occur at major clause boundaries (Shriberg, 1994), suggesting their presence may benefit parsing, we do not find this to be the case. Results shown for repair detection accuracy and its impact on parsing are consistent with previous work (Charniak and Johnson, 2001; Kahn et al., 2005; Harper et al., 2005).

Table 1: F-scores on Switchboard when trained in-domain. LB and Dep refer to relaxed labelled-bracket and dependency parse metrics (§3.2). Edit and filler word detection f-scores are also shown.

Edits	Fillers	Edit F	Filler F	LB	Dep
oracle	oracle	100.0	100.0	88.9	88.5
oracle	parser	100.0	93.1	87.8	87.9
parser	oracle	64.3	100.0	85.0	85.6
parser	parser	62.4	94.1	83.9	85.0

Our second set of experiments reports the effect of deleting fillers early when the parser is trained on text only (WSJ, §3.1). Our motivation here is to see if disfluency modeling, particularly filler detection, can help bleach speech data to more closely resemble text, thereby improving our ability to process it using text-based methods and training data (Rosenfeld et al., 1995). Again we contrast standard parsing with deleting disfluencies early (via oracle knowledge). Given our particular interest in fillers, we also report the effect of detecting them via a state-of-the-art system (Johnson et al., 2004).

Results appear in Table 2. It is worth noting that since our text-trained parser never produces FILLER or EDITED constituents, the bracket-based metric penalizes it for each such constituent appearing in the gold trees. Similarly, since the dependency metric ignores terminals occurring under these constituents in the gold trees, the metric penalizes the parser for producing dependencies for these termi-

Table 2: F-scores parsing Switchboard when trained on WSJ. Edit word detection varies between parser and oracle, and filler word detection varies between none, system (Johnson et al., 2004), and oracle. Filler F, LB, and Dep are defined as in Table 1.

Edits	Fillers	Filler F	LB	Dep
oracle	oracle	100.0	83.6	81.4
oracle	detect	89.3	81.6	80.5
oracle	none	-	71.8	75.4
none	oracle	100.0	76.3	76.7
none	detect	74.6	75.9	91.3
none	none	-	66.8	71.5

nals. Taken together, the two metrics provide a complementary perspective in interpreting results.

The trend observed across metrics and edit detection conditions shows that early deletion of system-detected fillers improves parsing accuracy 5-10%. As seen with in-domain training, early deletion of repairs is again seen to have a significant effect. Given that state-of-the-art edit detection performs at about 80% f-measure (Johnson and Charniak, 2004), much of the benefit derived here from oracle repair detection should be realizable in practice. The broader conclusion we draw from these results is that disfluency modeling has significant potential to improve text-based processing of speech data.

4 Conclusion

While early deletion of fillers has limited benefit for in-domain parsing of speech data, it can play an important role in *bleaching* speech data for more accurate text-based processing. Alternative methods of integrating detected filler information, such as parse reranking (Kahn et al., 2005), also merit investigation. It will also be important to evaluate the interaction with ASR error and sentence boundary detection error. In terms of bleaching, we saw that even with oracle detection of disfluency, our text-trained model still significantly under-performed the in-domain model, indicating additional methods for bleaching are still needed. We also plan to evaluating the benefit of disfluency modeling in bleaching speech data for text-based machine translation.

Acknowledgments

This work was supported by NSF grants 0121285, LIS9720368, and IIS0095940, and DARPA GALE contract HR0011-06-2-0001. We would like to thank Brian Roark, Mary Harper, and the rest of the JHU PASSED team for its support of this work.

References

E. Charniak and M. Johnson. 2001. Edit detection and parsing for transcribed speech. In *Proc. NAACL*, pages 118–126.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. NAACL*, pages 132–139.

M.G. Core and L.K. Schubert. 1999. A syntactic framework for speech repairs and other disruptions. In *Proc. ACL*, pages 413–420.

E. Black et al. 1991. Procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proc. Workshop on Speech and Natural Language*, pages 306–311.

D. Engel, E. Charniak, and M. Johnson. 2002. Parsing and disfluency placement. In *Proc. EMNLP*, pages 49–54.

D. Graff and S. Bird. 2000. Many uses, many annotations for large speech corpora: Switchboard and TDT as case studies. In *Proc. LREC*, pages 427–433.

M. Harper et al. *2005 Johns Hopkins Summer Workshop Final Report on Parsing and Spoken Structural Event Detection*.

J.G. Kahn et al. 2005. Effective use of prosody in parsing conversational speech. In *Proc. HLT/EMNLP*, 233–240.

M. Johnson and E. Charniak. 2004. A TAG-based noisy channel model of speech repairs. In *Proc. ACL*, pages 33–39.

M. Johnson, E. Charniak, and M. Lease. 2004. An improved model for recognizing disfluencies in conversational speech. In *Proc. Rich Text 2004 Fall Workshop (RT-04F)*.

D. Jones et al. 2003. Measuring the readability of automatic speech-to-text transcripts. In *Proc. Eurospeech*, 1585–1588.

Linguistic Data Consortium (LDC). 2004. Simple metadata annotation specification version 6.2.

M. Marcus et al. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2): 313–330.

B. Roark. 2002. Markov parsing: Lattice rescoring with a statistical parser. In *Proc. ACL*, pages 287–294.

R. Rosenfeld et al. 1995. Error analysis and disfluency modeling in the Swichboard domain: 1995 JHU Summer Workshop project team report.

E. Shriberg. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Ph.D. thesis, UC Berkeley.

A. Taylor, 1995. *Revision of Meteer et al.'s Dysfluency Annotation Stylebook for the Switchboard Corpus*. LDC.

A. Taylor, 1996. *Bracketing Switchboard: An addendum to the Treebank II Bracketing Guidelines*. LDC.

Evaluation of Utility of LSA for Word Sense Discrimination

Esther Levin
Dept. of Computer Science
City College of New York
NY, NY 10031
esther@cs.ccny.cuny.edu

Mehrbod Sharifi
Dept. of Computer Science
City College of New York
NY, NY 10031
mehrbod@yahoo.com

Jerry Ball
Air Force Research Laboratory
6030 S Kent Street
Mesa, AZ 85212-6061
Jerry.Ball@mesa.afmc.af.mil

Abstract

The goal of the on-going project described in this paper is evaluation of the utility of Latent Semantic Analysis (LSA) for unsupervised word sense discrimination. The hypothesis is that LSA can be used to compute context vectors for ambiguous words that can be clustered together – with each cluster corresponding to a different sense of the word. In this paper we report first experimental result on tightness, separation and purity of sense-based clusters as a function of vector space dimensionality and using different distance metrics.

1 Introduction

Latent semantic analysis (LSA) is a mathematical technique used in natural language processing for finding complex and hidden relations of meaning among words and the various contexts in which they are found (Landauer and Dumais, 1997; Landauer et al, 1998). LSA is based on the idea of association of elements (words) with contexts and similarity in word meaning is defined by similarity in shared contexts.

The starting point for LSA is the construction of a co-occurrence matrix, where the columns represent the different contexts in the corpus, and the rows the different word tokens. An entry *ij* in the matrix corresponds to the count of the number of times the word token *i* appeared in context *j*. Often the co-occurrence matrix is normalized for document length and word entropy (Dumais, 1994).

The critical step of the LSA algorithm is to compute the singular value decomposition (SVD) of the normalized co-occurrence matrix. If the matrices comprising the SVD are permuted such that the singular values are in decreasing order, they can be truncated to a much lower rank. According to Landauer and Dumais (1997), it is this dimensionality reduction step, the combining of surface information into a deeper abstraction that captures the mutual implications of words and passages and uncovers important structural aspects of a problem while filtering out noise. The singular vectors reflect principal components, or axes of greatest variance in the data, constituting the hidden abstract concepts of the semantic space, and each word and each document is represented as a linear combination of these concepts.

Within the LSA framework discreet entities such as words and documents are mapped into the same continuous low-dimensional parameter space, revealing the underlying semantic structure of these entities and making it especially efficient for variety of machine-learning algorithms. Following successful application of LSA to information retrieval other areas of application of the same methodology have been explored, including language modeling, word and document clustering, call routing and semantic inference for spoken interface control (Bellegarda, 2005).

The ultimate goal of the project described here is to explore the use of LSA for unsupervised identification of word senses and for estimating word sense frequencies from application relevant corpora following Schütze's (1998) context-group discrimination paradigm. In this paper we describe a first set of experiments investigating the tightness, separation and purity properties of sense-based clusters.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 77–80,
New York, June 2006.

2 Experimental Setup

The basic idea of the context-group discrimination paradigm adopted in this investigation is to induce senses of ambiguous word from their contextual similarity. The occurrences of an ambiguous word represented by their context vectors are grouped into clusters, where clusters consist of contextually similar occurrences. The context vectors in our experiments are LSA-based representation of the documents in which the ambiguous word appears. Context vectors from the training portion of the corpus are grouped into clusters and the centroid of the cluster—the sense vector—is computed. Ambiguous words from the test portion of the corpus are disambiguated by finding the closest sense vector (cluster centroid) to its context vector representation. If sense labels are available for the ambiguous words in the corpus, sense vectors are given a label that corresponds to the majority sense in their cluster, and sense discrimination accuracy can be evaluated by computing the percentage of ambiguous words from the test portion that were mapped to the sense vector whose label corresponds to the ambiguous word's sense label.

Our goal is to investigate how well the different senses of ambiguous words are separated in the LSA-based vector space. With an ideal representation the clusters of context vectors would be tight (the vectors in the cluster close to each other and close to centroid of the cluster), and far away from each other, and each cluster would be pure, i.e., consisting of vectors corresponding to words with the same sense. Since we don't want the evaluation of the LSA-based representation to be influenced by the choice of clustering algorithm, or the algorithm's initialization and its parameter settings that determine the resulting grouping, we took an orthogonal approach to the problem: Instead of evaluating the purity of the clusters based on geometrical position of vectors, we evaluate how well-formed the clusters based on sense labels are, how separated from each other and tight they are. As will be discussed below, performance evaluation of such sense-based clusters results in an upper bound on the performance that can be obtained by clustering algorithms such as EM or K-means.

3 Results

We used the line-hard-serve-interest corpus(Leacock et al, 1993), with 1151 instances for 3 noun senses of word "Line": cord - 373, division - 374, and text - 404; 752 instances for 2 adjective senses of word "Hard": difficult – 376, not yielding to pressure or easily penetrated – 376; 1292 instances for 2 verb senses of word "Serve": serving a purpose, role or function or acting as – 853, and providing service 439; and 2113 instances for 3 noun senses of word "Interest": readiness to give attention - 361, a share in a company or business – 500, money paid for the use of money -1252.

For all instances of an ambiguous word in the corpus we computed the corresponding LSA context vectors, and grouped them into clusters according to the sense label given in the corpus. To evaluate the inter-cluster tightness and intra-cluster separation for variable-dimensionality LSA representation we used the following measures:

1. Sense discrimination accuracy. To compute sense discrimination accuracy the centroid of each sense cluster was computed using 90% of the data. We evaluated the sense discrimination accuracy using the remaining 10% of the data reserved for testing by computing for each test context vector the closest cluster centroid and comparing their sense labels. To increase the robustness of this evaluation we repeated this computation 10 times, each time using a different 10% chunk for test data, round-robin style. The sense discrimination accuracy estimated in this way constitutes an upper bound on the sense discrimination performance of unsupervised clustering such as K-means or EM: The sense-based centroids, by definition, are the points with minimal average distance to all the same-sense points in the training set, while the centroids found by unsupervised clustering are based on geometric properties of all context vectors, regardless of their sense label.

2. Average Silhouette Value. The silhouette value (Rousseeuw, 1987) for each point is a measure of how similar that point is to points in its own cluster vs. points in other clusters. This measure ranges from +1, indicating points that are very distant from neighboring clusters, through 0, indicating points that are not distinctly in one cluster or another, to -1, indicating points that are probably assigned to the wrong cluster. To construct the silhouette value for each vector i, *S(i)*, the following formula is used:

$$S(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}},$$

where *a(i)* is an average distance of *i*-object to all other objects in the same cluster and *b(i)* is a minimum of average distance *of i*-object to all objects in other cluster (in other words, it is the average distance to the points in closest cluster among the other clusters). The overall average silhouette value is simply the average of the *S(i)* for all points in the whole dataset.

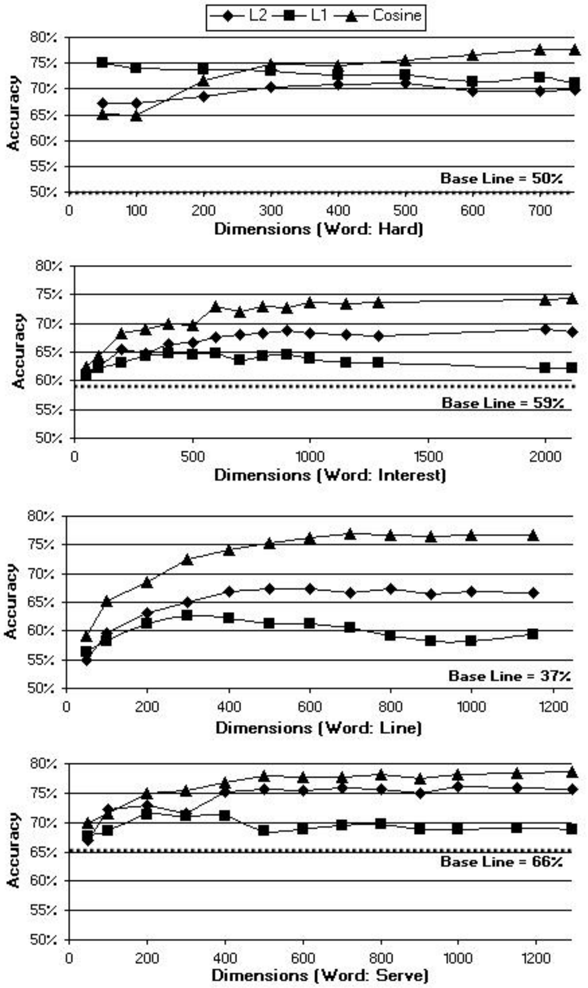


Figure 1: Average discrimination accuracy

Figure 1 plots the average discrimination accuracy as a function of LSA dimensionality for different distance/similarity measures, namely L2, L1 and cosine, for the 4 ambiguous words in the corpus. Note that the distance measure choice affects not only the classification of a point to the cluster, but also the computation of cluster centroid. For L2 and cosine measures the centroid is simply the average of vectors in the cluster, while for L1 it is the median, i.e., the value of *i*-th dimension of the cluster centroid vector is the median of values of the *i*-th dimension of all the vectors in the cluster.

As can be seen from the sense discrimination results in Fig. 1, cosine distance, the most frequently used distance measure in LSA applications, has the best performance in for 3 out of 4 words in the corpus. Only for "Hard" does L1 outperforms cosine for low values of LSA dimension. As to the influence of dimensionality reduction on sense discrimination accuracy, our results show that (at least for the cosine distance) the accuracy does not peak at any reduced dimension, rather it increases monotonically, first rapidly and then reaching saturation as the dimension is increased from its lowest value (50 in our experiments) to the full dimension that corresponds to the number of contexts in the corpus.

These results suggest that the value of dimensionality reduction is not in increasing the sense discrimination power of LSA representation, but in making the subsequent computations more efficient and perhaps enabling working with much larger corpora. For every number of dimensions examined, the average sense discrimination accuracy is significantly better than the baseline that was computed as the relative percentage of the most frequent sense of each ambiguous word in the corpus.

Figure 2 shows the average silhouette values for the sense-based clusters as a function of the dimensionality of the underlying LSA–based vector representation for the 3 different distance metrics and for the 4 words in the corpus. The average silhouette value is close to zero, not varying significantly for the different number of dimensions and distance measures. Although the measured silhouette values indicate that the sense-based clusters are not very tight, the sense-discrimination accuracy results suggest that they are sufficiently far from each other to guarantee relatively high accuracy.

4 Summary and Discussion

In this paper we reported on the first in a series of experiments aimed at examining the sense discrimination utility of LSA-based vector representation of ambiguous words' contexts. Our evaluation of average silhouette values indicates that sense-based clusters in the latent semantic space are not very tight (their silhouette values are mostly positive, but close to zero). However, they are separated enough to result in sense discrimination accuracy significantly higher than the baseline. We also found that the cosine distance measure outperforms L1 and L2, and that dimensionality reduction for sense-based clusters does not improve the sense discrimination accuracy.

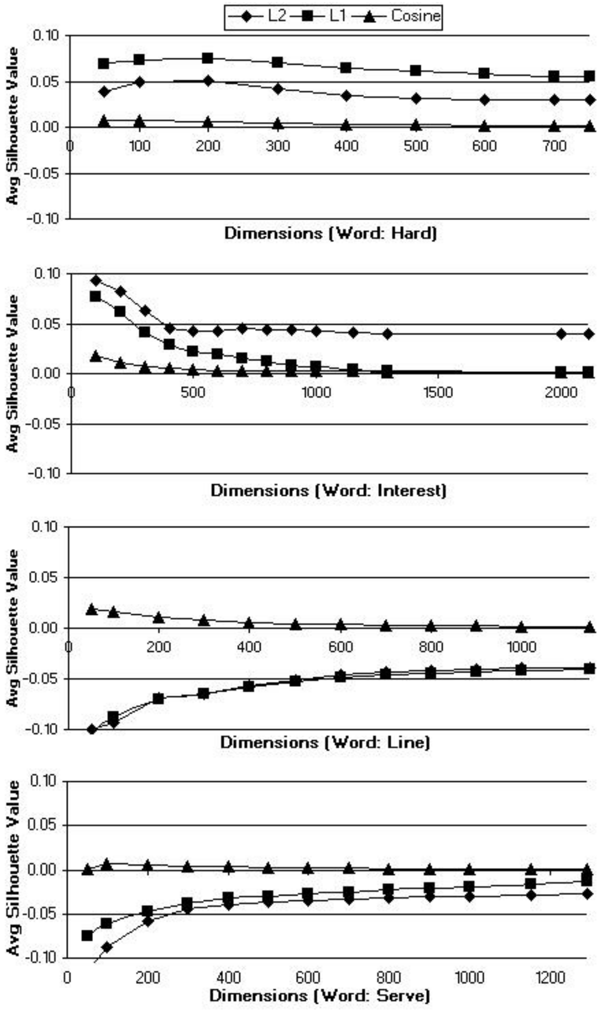


Figure2: Average silhouette values

The clustering examined in this paper is based on pre-established word sense labels, and the measured accuracy constitutes an upper bound on a sense discrimination accuracy that can be obtained by unsupervised clustering such as EM or segmental K-means. In the next phase of this investigation we plan to do a similar evaluation for clustering obtained without supervision by running K-means algorithm on the same corpus. Since such clustering is based on geometric properties of word vectors, we expect it to have a better tightness as measured by average silhouette value, but, at the same time, lower sense discrimination accuracy.

The experiments reported here are based on LSA representation computed using the whole document as a context for the ambiguous word. In the future we plan to investigate the influence of the context size on sense discrimination performance.

Acknowledgements

The project described above is supported by grant "Using LSA to Compute Word Sense Frequencies" from Air Force Research Lab. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the AFRL.

5 References

J.R. Bellegarda. 2005. *Latent Semantic Mapping*, IEEE Signal Processing Magazine, 22(5):70-80.

S.T. Dumais. 1994. *Latent Semantic Indexing (LSI) and TREC-2*, in Proc Second Text Retrieval Conf. (TREC-2), pp 104-105.

T.K. Landauer, S.T. Dumais. 1997. *A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge*, Psychological Review, 104(2):211-240.

T.K. Landauer, P. Foltz, and D. Laham. 1998. *Introduction to Latent Semantic Analysis*. Discourse Processes, 25, 259-284.

C. Leacock, G. Towel, E. Voorhees. 1993. *Corpus-Based Statistical Sense Resolution*, Proceedings of the ARPA Workshop on Human Language Technology.

P.J. Rousseeuw. 1987. *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*. Journal of Computational and Applied Mathematics. 20. 53-65.

H. Schütze. 1998. *Automatic Word Sense Discrimination*, Journal of Computational Linguistics, Volume 24, Number 2

Initial Study on Automatic Identification of Speaker Role in Broadcast News Speech

Yang Liu
University of Texas at Dallas, Richardson, TX
yangl@hlt.utdallas.edu

Abstract

Identifying a speaker's role (anchor, reporter, or guest speaker) is important for finding the structural information in broadcast news speech. We present an HMM-based approach and a maximum entropy model for speaker role labeling using Mandarin broadcast news speech. The algorithms achieve classification accuracy of about 80% (compared to the baseline of around 50%) using the human transcriptions and manually labeled speaker turns. We found that the maximum entropy model performs slightly better than the HMM, and that the combination of them outperforms any model alone. The impact of the contextual role information is also examined in this study.

1 Introduction

More effective information access is beneficial to deal with the increasing amount of broadcast news speech. Many attempts have been made in the past decade to build news browser, spoken document retrieval system, and summarization or question answering system to effectively handle the large volume of news broadcast speech (e.g., the recent DARPA GALE program). Structural information, such as story segmentation or speaker clustering, is critical for all of these applications. In this paper, we investigate automatic identification of the speakers' roles in broadcast news speech. A speaker's role (such as anchor, reporter or journalist, interviewee, or some soundbites) can provide useful structural information of broadcast news. For example, anchors appear through the entire program and generally introduce news stories. Reporters typically report a specific news story, in which there may be other guest speakers. The transition between anchors and reporters is usually a good indicator of story structure. Speaker role information was shown to be useful for summarizing broadcast news (Maskey and Hirschberg, 2003). Anchor information has also been used for video segmentation, such as the systems in the TRECVID evaluations.[1]

In this paper, we develop algorithms for speaker role identification in broadcast news speech. Human transcription and manual speaker turn labels are used in this initial study. The task is then to classify each speaker's turn as *anchor*, *reporter*, or *other*. We use about 170 hours of speech for training and testing. Two approaches are evaluated, an HMM and a maximum entropy classifier. Our methods achieve about 80% accuracy for the three-way classification task, compared to around 50% when every speaker is labeled with the majority class label, i.e., anchor.[2]

The rest of the paper is organized as follows. Related work is introduced in Section 2. We describe our approaches in Section 3. Experimental setup and results are presented in Section 4. Summary and future work appear in Section 5.

2 Related Work

The most related previous work is (Barzilay et al., 2000), in which Barzilay et al. used BoosTexter and the maximum entropy model to classify each speaker's role in an English broadcast news corpus. Three classes are used, anchor, journalist, and guest speaker, which are very similar to the role categories in our study. Lexical features (key words), context features, duration, and explicit speaker introduction are used as features. For the three-way classification task, they reported accuracy of about 80% compared to the chance of 35%. They have investigated using both the reference transcripts and speech recognition output. Our study differs from theirs in that we use one generative modeling approach (HMM), as well as the conditional maximum entropy method. We also evaluate the contextual role information for classification. In addition, our experiments are conducted using a different language, Mandarin broadcast news. There may be inherent difference across languages and news sources.

Another task related to our study is anchor segmentation. Huang et al. (Huang et al., 1999) used a recognition model for a particular anchor and a background model to identify anchor segments. They reported very promising results for the task of determining whether

[1] See http://www-nlpir.nist.gov/projects/trecvid/ for more information on video retrieval evaluations.

[2] Even though this is a baseline (or chance performance), it is not very meaningful since there is no information provided in this output.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 81–84,
New York, June 2006. ©2006 Association for Computational Linguistics

or not a particular anchor is talking. However, this method is not generalizable to multiple anchors, nor is it to reporters or other guest speakers. Speaker role detection is also related to speaker segmentation and clustering (also called speaker diarization), which was a benchmark test in the NIST Rich Transcription evaluations in the past few years (for example, NIST RT-04F http://www.nist.gov/speech/tests/rt/rt2004/fall/). Most of the speaker diarization systems only use acoustic information; however, in recent studies textual sources have also been utilized to help improve speaker clustering results, such as (Canseco et al., 2005). The goal of speaker diarization is to identify speaker change and group the same speakers together. It is different from our task since we determine the role of a speaker rather than speaker identity. In this initial study, instead of using automatic speaker segmentation and clustering results, we use the manual speaker segments but without any speaker identity information.

3 Speaker Role Identification Approaches

3.1 Hidden Markov Model (HMM)

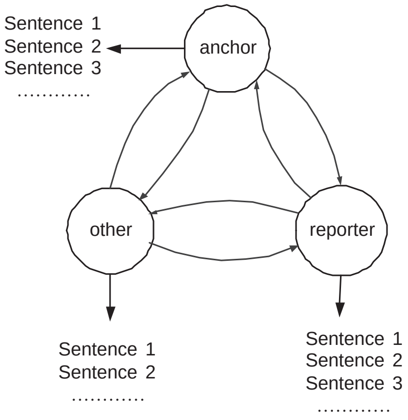


Figure 1: A graphical representation of the HMM approach for speaker role labeling. This is a simple first order HMM.

The HMM has been widely used in many tagging problems. Stolcke et al. (Stolcke et al., 2000) used it for dialog act classification, where each utterance (or dialog act) is used as the observation. In speaker role detection, the observation is composed of a much longer word sequence, i.e., the entire speech from one speaker. Figure 1 shows the graphical representation of the HMM for speaker role identification, in which the states are the speaker roles, and the observation associated with a state consists of the utterances from a speaker. The most likely role sequence $\hat{R}$ is:

$$\hat{R} = \operatorname*{argmax}_R P(R|O) = \operatorname*{argmax}_R P(O|R)P(R), \quad (1)$$

where O is the observation sequence, in which O_i corresponds to one speaker turn. If we assume what a speaker says is only dependent on his or her role, then:

$$P(O|R) = \prod_i P(O_i|R_i). \quad (2)$$

From the labeled training set, we train a language model (LM), which provides the transition probabilities in the HMM, i.e., the $P(R)$ term in Equation (1). The vocabulary in this role LM (or role grammar) consists of different role tags. All the sentences belonging to the same role are put together to train a role specific word-based N-gram LM. During testing, to obtain the observation probabilities in the HMM, $P(O_i|R_i)$, each role specific LM is used to calculate the perplexity of those sentences corresponding to a test speaker turn.

The graph in Figure 1 is a first-order HMM, in which the role state is only dependent on the previous state. In order to capture longer dependency relationship, we used a 6-gram LM for the role LM. For each role specific word-based LM, 4-gram is used with Kneser-Ney smoothing. There is a weighting factor when combining the state transitions and the observation probabilities with the best weights tuned on the development set (6 for the transition probabilities in our experiments). In addition, in stead of using Viterbi decoding, we used forward-backward decoding in order to find the most likely role tag for each segment. Finally we may use only a subset of the sentences in a speaker's turn, which are possibly more discriminative to determine the speaker's role. The LM training and testing and HMM decoding are implemented using the SRILM toolkit (Stolcke, 2002).

3.2 Maximum Entropy (Maxent) Classifier

A Maxent model estimates the conditional probability:

$$P(R_i|O) = \frac{1}{Z_\lambda(O)} exp(\sum_k \lambda_k g_k(R_i, O)), \quad (3)$$

where $Z_\lambda(O)$ is the normalization term, functions $g_k(R_i, O)$ are indicator functions weighted by λ, and k is used to indicate different 'features'. The weights (λ) are obtained to maximize the conditional likelihood of the training data, or in other words, maximize the entropy while satisfying all the constraints. Gaussian smoothing (variance=1) is used to avoid overfitting. In our experiments we used an existing Maxent toolkit (available from http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html).

The following features are used in the Maxent model:

- bigram and trigram of the words in the first and the last sentence of the current speaker turn
- bigram and trigram of the words in the last sentence of the previous turn

- bigram and trigram of the words in the first sentence of the following turn

Our hypothesis is that the first and the last sentence from a speaker's turn are more indicative of the speaker's role (e.g., self introduction and closing). Similarly the last sentence from the previous speaker segment and the first sentence of the following speaker turn also capture the speaker transition information. Even though sentences from the other speakers are included as features, the Maxent model makes a decision for each test speaker turn individually without considering the other segments. The impact of the contextual role tags will be evaluated in our experiments.

4 Experiments

4.1 Experimental Setup

We used the TDT4 Mandarin broadcast news data in this study. The data set consists of about 170 hours (336 shows) of news speech from different sources. In the original transcripts provided by LDC, stories are segmented; however, speaker information (segmentation or identity) is not provided. Using the reference transcripts and the audio files, we manually labeled the data with speaker turns and the role tag for each turn.[3] Speaker segmentation is generally very reliable; however, the role annotation is ambiguous in some cases. The interannotator agreement will be evaluated in our future work. In this initial study, we just treat the data as noisy data.

We preprocessed the transcriptions by removing some bad codes and also did text normalization. We used punctuation (period, question mark, and exclamation) available from the transcriptions (though not very accurate) to generate sentences, and a left-to-right longest word match approach to segment sentences into words. These words/sentences are then used for feature extraction in the Maxent model, and LM training and perplexity calculation in the HMM as described in Section 3. Note that the word segmentation approach we used may not be the-state-of-art, which might have some effect on our experiments.

10-fold cross validation is used in our experiments. The entire data set is split into ten subsets. Each time one subset is used as the test set, another one is used as the development set, and the rest are used for training. The average number of segments (i.e., speaker turns) in the ten subsets is 1591, among which 50.8% are anchors. Parameters (e.g., weighting factor) are tuned based on the average performance over the ten development sets, and the same weights are applied to all the splits during testing.

4.2 Results

A **HMM and Maxent**: Table 1 shows the role identification results using the HMM and the Maxent model, including the overall classification accuracy and the precision/recall rate (%) for each role. These results are the average over the 10 test sets.

	HMM		Maxent	
	precision	recall	precision	recall
anchor	78.03	87.33	80.29	87.23
reporter	78.54	66.42	73.34	77.01
other	83.05	68.19	89.52	41.30
Accuracy (%)	77.18		77.42	

Table 1: Automatic role labeling results (%) using the HMM and Maxent classifiers.

From Table 1 we find that the overall classification performance is similar when using the HMM and the Maxent model; however, their error patterns are quite different. For example, the Maxent model is better than the HMM at identifying "reporter" role, but worse at identifying "other" speakers (see the recall rate shown in the table). In the HMM, we only used the first and the last sentence in a speaker's turn, which are more indicative of the speaker's role. We observed significant performance degradation, that is, 74.68% when using all the sentences for LM training and perplexity calculation, compared to 77.18% as shown in the table using a subset of a speaker's speech. Note that the sentences used in the HMM and Maxent models are the same; however, the Maxent does not use any contextual role tags (which we will examine next), although it does include some words from the previous and the following speaker segments in its feature set.

B **Contextual role information**: In order to investigate how important the role sequence is, we conducted two experiments for the Maxent model. In the first experiment, for each segment, the reference role tag of the previous and the following segments and the combination of them are included as features for model training and testing (a "cheating" experiment). In the second experiment, a two-step approach is employed. Following the HMM and Maxent experiments (i.e., results as shown in Table 1), Viterbi decoding is performed using the posterior probabilities from the Maxent model and the transition probabilities from the role LM as in the HMM (with weight 0.3). The average performance over the ten test sets is shown in Table 2 for these two experiments. For comparison, we also present the decoding results of the HMM with and without using sequence information (i.e., the transition probabilities in the HMM). Additionally, the system combination

[3] The labeling guideline can be found from http://www.hlt.utdallas.edu/˜yangl/spkr-label/. It was modified based on the annotation manual used for English at Columbia University (available from http://www1.cs.columbia.edu/˜smaskey/labeling/Labeling_Manual_v_2_1.pdf).

results of the HMM and Maxent are presented in the table, with more discussion on this later. We observe from Table 2 that adding contextual role information improves performance. Including the two reference role tags yields significant gain in the Maxent model, even though some sentences from the previous and the following segments are already included as features. The HMM suffers more than the Maxent classifier when role sequence information is not used during decoding, since that is the only contextual information used in the HMM, unlike the Maxent model, which uses features extracted from the neighboring speaker turns.

	Accuracy (%)
0: Maxent (as in Table 1)	77.42
1: Maxent + 2 reference tags	80.90
2: Maxent + sequence decoding	78.59
3: HMM (as in Table 1)	77.18
4: HMM w/o sequence	73.30
Maxent (0) + HMM (3)	79.74
Maxent (2) + HMM (3)	81.97

Table 2: Impact of role sequence information on the HMM and Maxent classifiers. The combination results of the HMM and Maxent are also provided.

C **System combination**: For system combination, we used two different Maxent results: with and without the Viterbi sequence decoding, corresponding to experiments (0) and (2) as shown in Table 2 respectively. When combining the HMM and Maxent, i.e., the last two rows in Table 2, the posterior probabilities from them are linearly weighted (weight 0.6 for the Maxent in the upper one, and 0.7 for the Maxent in the bottom one). The combination of the two approaches yields better performance than any single model in the two cases. We also investigated other system combination approaches. For example, a decision tree or SVM that builds a 3-way superclassifier using the posterior probabilities from the HMM and Maxent. However, so far we have not found any gain from more complicated system combination than a simple linear interpolation. We will study this in our future work.

5 Summary and Future Work

In this paper we have reported an initial study of speaker role identification in Mandarin broadcast news speech using the HMM and Maxent tagging approaches. We find that the conditional Maxent generally performs slightly better than the HMM, and that their combination outperforms each model alone. The HMM and the Maxent model show differences in identifying different roles. The impact of contextual role information is also examined for the two approaches, and a significant gain is observed when contextual information is modeled. We find that the beginning and the end sentences in a speaker's turn are good cues for role identification. The overall classification performance in this study is similar to that reported in (Barzilay et al., 2000); however, the chance performance is quite different (35% in that study). It is not clear yet whether it is because of the difference across the two corpora or languages.

The Maxent model provides a convenient way to incorporate various knowledge sources. We will investigate other features to improve the classification results, such as name information, acoustic or prosodic features, and speaker clustering results (considering that the same speaker typically has the same role tag). We plan to examine the effect of using speech recognition output, as well as automatic speaker segmentation and clustering results. Analysis of difference news sources may also reveal some interesting findings. Since our working hypothesis is that speaker role information is important to find structure in broadcast news, we will investigate whether and how speaker role relates to downstream language processing applications, such as summarization or question answering.

Acknowledgment

The author thanks Julia Hirschberg and Sameer Maskey at Columbia University and Mari Ostendorf at the University of Washington for the useful discussions, and Mei-Yuh Hwang for helping with Mandarin word segmentation and text normalization. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA.

References

R. Barzilay, M. Collins, J. Hirschberg, and S. Whittaker. 2000. The rules behind roles: Identifying speaker role in radio broadcasts. In *Proc. of AAAI*.

L. Canseco, L. Lamel, and J Gauvain. 2005. A comparative study using manual and automatic transcription for diarization. In *Proc. of ASRU*.

Q. Huang, Z. Liu, A. Rosenberg, D. Gibbon, and B. Shahraray. 1999. Automated generation of news content hierarchy by integrating audio, video, and text information. In *Proc. of ICASSP*, pages 3025–3028.

S. Maskey and J. Hirschberg. 2003. Automatic summarization of broadcast news using structural features. In *Eurospeech*.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurasky, P. Taylor, R. Martin, C.V. Ess-Dykema, and M. Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

A. Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proc. of ICSLP*, pages 901–904.

Automatic Recognition of Personality in Conversation

François Mairesse
Department of Computer Science
University of Sheffield
Sheffield, S1 4DP, United Kingdom
F.Mairesse@sheffield.ac.uk

Marilyn Walker
Department of Computer Science
University of Sheffield
Sheffield, S1 4DP, United Kingdom
M.A.Walker@sheffield.ac.uk

Abstract

The identification of personality by automatic analysis of conversation has many applications in natural language processing, from leader identification in meetings to partner matching on dating websites. We automatically train models of the main five personality dimensions, on a corpus of conversation extracts and personality ratings. Results show that the models perform better than the baseline, and their analysis confirms previous findings linking language and personality, while revealing many new linguistic and prosodic markers.

1 Introduction

It is well known that utterances convey information about the *speaker* in addition to their semantic content. One such type of information consists of cues to the speaker's *personality traits*, typically assessed along five dimensions known as the Big Five (Norman, 1963):

- Extraversion (sociability, assertiveness)
- Emotional stability (vs. neuroticism)
- Agreeableness to other people (friendliness)
- Conscientiousness (discipline)
- Intellect (openness to experience)

Findings include that extraverts talk more, louder, and faster, with fewer pauses and hesitations, and more informal language (Scherer, 1979; Furnham, 1990; Heylighen and Dewaele, 2002; Gill and Oberlander, 2002). Neurotics use more 1st person singular pronouns and negative emotion words, while conscientious people avoid negations and negative emotion words (Pennebaker and King, 1999). The use of words related to insight and the avoidance of past tense indicate intellect, and swearing and negative emotion words mark disagreeableness. Correlations are higher in spoken language, possibly especially in informal conversation (Mehl et al., in press).

Previous work has modeled emotion and personality in virtual agents, and classified emotions from actor's speech (André et al., 1999; Liscombe et al., 2003). However, to our knowledge no one has tested whether it is possible to automatically recognize personality from conversation extracts of unseen subjects. Our hypothesis is that automatic analysis of conversation to detect personality has application in a wide range of language processing domains. Identification of leaders using personality dimensions could be useful in analyzing meetings and the conversations of suspected terrorists (Hogan et al., 1994; Tucker and Whittaker, 2004; Nunn, 2005). Dating websites could analyze text messages to try to match personalities and increase the chances of a successful relationship (Donnellan et al., 2004). Dialogue systems could adapt to the user's personality, like humans do (Reeves and Nass, 1996; Funder and Sneed, 1993). This work is a first step toward individual adaptation in dialogue systems.

We present non-linear statistical models for ranking utterances based on the Big Five personality traits. Results show that the models perform significantly better than a random baseline, and that prosodic features are good indicators of extraversion. A qualitative analysis confirms previous findings linking language and personality, while revealing many new linguistic markers.

2 Experimental method

Our approach can be summarized in five steps: (1) collect individual corpora; (2) collect personality

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 85–88,
New York, June 2006.

ratings for each participant; (3) extract relevant features from the texts; (4) build statistical models of the personality ratings based on the features; and (5) test the learned models on the linguistic outputs of unseen individuals.

2.1 Spoken language and personality ratings

The data consists of daily-life conversation extracts of 96 participants wearing an Electronically Activated Recorder (EAR) for two days, collected by Mehl et al. (in press). To preserve the participants' privacy, random bits of conversation were recorded, and only the *participants'* utterances were transcribed, making it impossible to reconstruct whole conversations. The corpus contains 97,468 words and 15,269 utterances. Table 1 shows utterances for two participants judged as introvert and extravert.

Introvert:
- Yeah you would do kilograms. Yeah I see what you're saying. - On Tuesday I have class. I don't know. - I don't know. A16. Yeah, that is kind of cool. - I don't know. I just can't wait to be with you and not have to do this every night, you know? - Yeah. You don't know. Is there a bed in there? Well ok just...
Extravert:
- That's my first yogurt experience here. Really watery. Why? - Damn. New game. - Oh. - Yeah, but he, they like each other. He likes her. - They are going to end up breaking up and he's going to be like.

Table 1: Extracts from the corpus, for participants rated as extremely introvert and extravert.

Between 5 and 7 independent observers scored each extract using the Big Five Inventory (John and Srivastava, 1999). Mehl et al. (in press) report strong inter-observer reliabilities for all dimensions ($r = 0.84$, $p < 0.01$). Average observers' ratings were used as the scores for our experiments.

2.2 Feature selection

Features are *automatically* extracted from each extract (see Table 2). We compute the ratio of words in each category from the LIWC utility (Pennebaker et al., 2001), as those features are correlated with the Big Five dimensions (Pennebaker and King, 1999). Additional psychological characteristics were computed by averaging word feature counts from the MRC psycholinguistic database (Coltheart, 1981). In an attempt to capture initiative-taking in conversation (Walker and Whittaker, 1990; Furnham, 1990), we introduce utterance type features using heuristics on the parse tree to tag each utterance as a command, prompt, question or assertion. Overall tagging accuracy over 100 randomly selected utterances is 88%. As personality influences speech, we also use Praat (Boersma, 2001) to compute prosodic features characterizing the voice's pitch, intensity, and speech rate.

LIWC FEATURES (Pennebaker et al., 2001):

· STANDARD COUNTS:
- Word count (WC), words per sentence (WPS), type/token ratio (Unique), words captured (Dic), words longer than 6 letters (Sixltr), negations (Negate), assents (Assent), articles (Article), prepositions (Preps), numbers (Number)
- Pronouns (Pronoun): 1st person singular (I), 1st person plural (We), total 1st person (Self), total 2nd person (You), total 3rd person (Other)

· PSYCHOLOGICAL PROCESSES:
- Affective or emotional processes (Affect): positive emotions (Posemo), positive feelings (Posfeel), optimism and energy (Optim), negative emotions (Negemo), anxiety or fear (Anx), anger (Anger), sadness (Sad)
- Cognitive Processes (Cogmech): causation (Cause), insight (Insight), discrepancy (Discrep), inhibition (Inhib), tentative (Tentat), certainty (Certain)
- Sensory and perceptual processes (Senses): seeing (See), hearing (Hear), feeling (Feel)
- Social processes (Social): communication (Comm), other references to people (Othref), friends (Friends), family (Family), humans (Humans)

· RELATIVITY:
- Time (Time), past tense verb (Past), present tense verb (Present), future tense verb (Future)
- Space (Space): up (Up), down (Down), inclusive (Incl), exclusive (Excl)
- Motion (Motion)

· PERSONAL CONCERNS:
- Occupation (Occup): school (School), work and job (Job), achievement (Achieve)
- Leisure activity (Leisure): home (Home), sports (Sports), television and movies (TV), music (Music)
- Money and financial issues (Money)
- Metaphysical issues (Metaph): religion (Relig), death (Death), physical states and functions (Physcal), body states and symptoms (Body), sexuality (Sexual), eating and drinking (Eating), sleeping (Sleep), grooming (Groom)

· OTHER DIMENSIONS:
- Punctuation (Allpct): period (Period), comma (Comma), colon (Colon), semi-colon (Semic), question (Qmark), exclamation (Exclam), dash (Dash), quote (Quote), apostrophe (Apostro), parenthesis (Parenth), other (Otherp)
- Swear words (Swear), nonfluencies (Nonfl), fillers (Fillers)

MRC FEATURES (Coltheart, 1981):

Number of letters (Nlet), phonemes (Nphon), syllables (Nsyl), Kucera-Francis written frequency (K-F-freq), Kucera-Francis number of categories (K-F-ncats), Kucera-Francis number of samples (K-F-nsamp), Thorndike-Lorge written frequency (T-L-freql), Brown verbal frequency (Brown-freq), familiarity rating (Fam), concreteness rating (Conc), imageability rating (Imag), meaningfulness Colorado Norms (Meanc), meaningfulness Paivio Norms (Meanp), age of acquisition (AOA)

UTTERANCE TYPE FEATURES:

Ratio of commands (Command), prompts or back-channels (Prompt), questions (Question), assertions (Assertion)

PROSODIC FEATURES:

Average, minimum, maximum and standard deviation of the voice's pitch in Hz (Pitch-mean, Pitch-min, Pitch-max, Pitch-stddev) and intensity in dB (Int-mean, Int-min, Int-max, Int-stddev), voiced time (Voiced) and speech rate (Word-per-sec)

Table 2: Description of all features, with feature labels in brackets.

2.3 Statistical model

By definition, personality evaluation assesses relative differences between individuals, e.g. one person is described as an extravert because the average population is not. Thus, we formulate personality recognition as a ranking problem: given two individuals' extracts, which shows more extraversion?

Personality models are trained using RankBoost, a boosting algorithm for ranking, for each Big Five trait using the observers' ratings of personality (Freund et al., 1998). RankBoost expresses the learned models as rules, which support the analysis of differences in the personality models (see section 3). Each rule modifies the conversation extract's ranking score by α whenever a feature value exceeds experimentally learned thresholds, e.g. Rule 1 of the extraversion model in Table 4 increases the score of an extract by $\alpha = 1.43$ if the speech rate is above 0.73 words per second. Models are evaluated by a ranking error function which reports the percentage of misordered pairs of conversation extracts.

3 Results

The features characterize many aspects of language production: utterance types, content and syntax (LIWC), psycholinguistic statistics (MRC), and prosody. To evaluate how each feature set contributes to the final result, we trained models with the full feature set and with each set individually. Results are summarized in Table 3. The baseline is a model ranking extracts randomly, producing a ranking error of 0.5 on average. Results are averaged over a 10 fold cross-validation.

Feature set	All	LIWC	MRC	Type	Pros
Set size	117	88	14	4	11
Extraversion	0.35•	0.36•	0.45	0.55	**0.26•**
Emot. stability	0.40	0.41	**0.39•**	0.43	0.45
Agreeableness	**0.31•**	0.32•	0.44	0.45	0.54
Conscientious.	**0.33•**	0.36•	0.41•	0.44	0.55
Intellect	0.38•	**0.37•**	0.41	0.49	0.44

• statistically significant improvement over the random ordering baseline (two-tailed paired t-test, $p < 0.05$)

Table 3: Ranking errors over a 10 fold cross-validation for different feature sets (Type=utterance type, Pros=prosody). Best models are in bold.

Paired t-tests show that models of extraversion, agreeableness, conscientiousness and intellect using all features are better than the random ordering baseline (two-tailed, $p < 0.05$)[1]. Emotional stability is the most difficult trait to model, while agreeableness and conscientiousness produce the best results, with ranking errors of 0.31 and 0.33 respectively. Table 3 shows that LIWC features perform significantly better than the baseline for all dimensions but emotional stability, while emotional stability is best predicted by MRC features. Interestingly, prosodic features are very good predictors of extraversion, with a lower ranking error than the full feature set (0.26), while utterance type features on their own never outperform the baseline.

The RankBoost rules indicate the impact of each feature on the recognition of a personality trait by the magnitude of the parameter α associated with that feature. Table 4 shows the rules with the most impact on each best model, with the associated α values. The feature labels are in Table 2. For example, the model of extraversion confirms previous findings by associating this trait with a high speech rate (Rules 1 and 4) and longer conversations (Rule 5). But many new markers emerge: extraverts speak with a high pitch (Rules 2, 6 and 7), while introverts' pitch varies a lot (Rules 15, 18 and 20). Agreeable people use longer words but shorter sentences (Rule 1 and 20), while swear words reduce the agreeableness score (Rules 12, 18 and 19). As expected, conscientious people talk a lot about their job (Rule 1), while unconscientious people swear a lot and speak loudly (Rules 19 and 20). Our models contain many additional personality cues which aren't identified through a typical correlational analysis.

4 Conclusion

We showed that personality can be recognized automatically in conversation. To our knowledge, this is the first report of experiments testing trained models on unseen subjects. There are models for each dimension that perform significantly better than the baseline. Combinations of these models may be useful to identify important personality types in different NLP applications, e.g. a combination of extraversion, emotional stability and intellect indicates leadership, while low intellect, extraversion and agreeableness are correlated with perceptions of trustworthiness.

One limitation for applications involving speech recognition is that recognition errors will introduce noise in all features except prosodic features, and prosodic features on their own are only effective in the extraversion model. However, our data set is relatively small (96 subjects) so we expect that more

[1] We also built models of self-reports of personality, but none of them significantly outperforms the baseline.

#	**Extraversion**		**Emotional stability**		**Agreeableness**		**Conscientiousness**		**Intellect**	
	with prosody	α	**with MRC**	α	**with all**	α	**with all**	α	**with LIWC**	α
1	Word-per-sec ≥ 0.73	1.43	Nlet ≥ 3.28	0.53	Nphon ≥ 2.66	0.56	Occup ≥ 1.21	0.37	Colon ≥ 0.03	0.49
2	Pitch-mean ≥ 194.61	0.41	T-L-freq ≥ 28416	0.25	Tentat ≥ 2.83	0.50	Insight ≥ 2.15	0.36	Insight ≥ 1.75	0.37
3	Voiced ≥ 647.35	0.41	Meanc ≥ 384.17	0.24	Colon ≥ 0.03	0.41	Posfeel ≥ 0.30	0.30	Job ≥ 0.29	0.33
4	Word-per-sec ≥ 2.22	0.36	AOA ≥ 277.36	0.24	Posemo ≥ 2.67	0.32	Int-stddev ≥ 7.83	0.29	Music ≥ 0.18	0.32
5	Voiced ≥ 442.95	0.31	K-F-nsamp ≥ 322	0.22	Voiced ≥ 584	0.32	Nlet ≥ 3.29	0.27	Optim ≥ 0.19	0.24
6	Pitch-max ≥ 599.88	0.30	Meanp ≥ 654.57	0.19	Relig ≥ 0.43	0.27	Comm ≥ 1.20	0.26	Inhib ≥ 0.15	0.24
7	Pitch-mean ≥ 238.99	0.26	Conc ≥ 313.55	0.17	Insight ≥ 2.09	0.25	Nphon ≥ 2.66	0.25	Tentat ≥ 2.23	0.22
8	Int-stddev ≥ 6.96	0.24	K-F-ncats ≥ 14.08	0.15	Prompt ≥ 0.06	0.25	Nphon ≥ 2.67	0.22	Posemo ≥ 2.67	0.19
9	Int-max ≥ 85.87	0.24	Nlet ≥ 3.28	0.14	Comma ≥ 4.60	0.23	Nphon ≥ 2.76	0.20	Future ≥ 0.87	0.17
10	Voiced ≥ 132.35	0.23	Nphon ≥ 2.64	0.13	Money ≥ 0.38	0.20	K-F-nsamp ≥ 329	0.19	Certain ≥ 0.92	0.17
11	Pitch-max ≥ 636.35	-0.05	Fam ≥ 601.98	-0.19	Fam ≥ 601.61	-0.16	Swear ≥ 0.20	-0.18	Affect ≥ 5.07	-0.16
12	Pitch-slope ≥ 312.67	-0.06	Nphon ≥ 2.71	-0.19	Swear ≥ 0.41	-0.18	WPS ≥ 6.25	-0.19	Achieve ≥ 0.62	-0.17
13	Int-min ≥ 54.30	-0.06	AOA ≥ 308.39	-0.23	Anger ≥ 0.92	-0.19	Pitch-mean ≥ 229	-0.20	Othref ≥ 7.67	-0.17
14	Word-per-sec ≥ 1.69	-0.06	Brown-freq ≥ 1884	-0.25	Time ≥ 3.71	-0.20	Othref ≥ 7.64	-0.20	I ≥ 7.11	-0.19
15	Pitch-stddev ≥ 115.49	-0.06	Fam ≥ 601.07	-0.25	Negate ≥ 3.52	-0.20	Humans ≥ 0.83	-0.21	WPS ≥ 5.60	-0.20
16	Pitch-max ≥ 637.27	-0.06	K-F-nsamp ≥ 329	-0.26	Fillers ≥ 0.54	-0.22	Swear ≥ 0.93	-0.21	Social ≥ 10.56	-0.20
17	Pitch-slope ≥ 260.51	-0.12	Imag ≥ 333.50	-0.27	Time ≥ 3.69	-0.23	Swear ≥ 0.17	-0.24	You ≥ 3.57	-0.21
18	Pitch-stddev ≥ 118.10	-0.15	Meanp ≥ 642.81	-0.28	Swear ≥ 0.61	-0.27	Relig ≥ 0.32	-0.27	Incl ≥ 4.30	-0.33
19	Int-stddev ≥ 6.30	-0.18	K-F-ncats ≥ 14.32	-0.35	Swear ≥ 0.45	-0.27	Swear ≥ 0.65	-0.31	Physcal ≥ 1.79	-0.33
20	Pitch-stddev ≥ 119.73	-0.47	Nsyl ≥ 1.17	-0.63	WPS ≥ 6.13	-0.45	Int-max ≥ 86.84	-0.50	Family ≥ 0.08	-0.39

Table 4: Best RankBoost models for each trait. Rows 1-10 represent the rules producing the highest score increase, while rows 11-20 indicate evidence for the other end of the scale, e.g. introversion.

training data would improve model accuracies and might also make additional features useful. In future work, we plan to integrate these models in a dialogue system to adapt the system's language generation; we will then be able to test whether the accuracies we achieve are sufficient and explore methods for improving them.

Acknowledgements

Thanks to Matthias Mehl and James Pennebaker for sharing their data.

References

E. André, M. Klesen, P. Gebhard, S. Allen, and T. Rist. 1999. Integrating models of personality and emotions into lifelike characters. In *Proc. of the International Workshop on Affect in Interactions*, p. 136–149.

P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345.

M. Coltheart. 1981. The MRC psycholinguistic database. *Quarterly J. of Experimental Psychology*, 33A:497–505.

B. Donnellan, R. D. Conger, and C. M. Bryant. 2004. The Big Five and enduring marriages. *J. of Research in Personality*, 38:481–504.

Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. 1998. An efficient boosting algorithm for combining preferences. In *Proc. of the 15th ICML*, p. 170–178.

D. Funder and C. Sneed. 1993. Behavioral manifestations of personality: An ecological approach to judgmental accuracy. *J. of Personality and Social Psychology*, 64(3):479–490.

A. Furnham, 1990. *Handbook of Language and Social Psychology*, chapter Language and Personality. Winley.

A. J. Gill and J. Oberlander. 2002. Taking care of the linguistic features of extraversion. In *Proc. of the 24th Annual Conference of the Cognitive Science Society*, p. 363–368.

F. Heylighen and J.-M. Dewaele. 2002. Variation in the contextuality of language: an empirical measure. *Context in Context, Special issue of Foundations of Science*, 7:293–340.

R. Hogan, G. J. Curphy, and J. Hogan. 1994. What we know about leadership: Effectiveness and personality. *American Psychologist*, 49(6):493–504.

O. P. John and S. Srivastava. 1999. The Big Five trait taxonomy: History, measurement, and theoretical perspectives. In L. A. Pervin and O. P. John, editors, *Handbook of personality theory and research*. New York: Guilford Press.

J. Liscombe, J. Venditti, and J. Hirschberg. 2003. Classifying subject ratings of emotional speech using acoustic features. In *Proc. of Eurospeech - Interspeech 2003*, p. 725–728.

M. R. Mehl, S. D. Gosling, and J. W. Pennebaker. In press. Personality in its natural habitat: Manifestations and implicit folk theories of personality in daily life. *J. of Personality and Social Psychology*.

W. T. Norman. 1963. Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality rating. *J. of Abnormal and Social Psychology*, 66:574–583.

S. Nunn. 2005. Preventing the next terrorist attack: The theory and practice of homeland security information systems. *J. of Homeland Security and Emergency Management*, 2(3).

J. W. Pennebaker and L. A. King. 1999. Linguistic styles: Language use as an individual difference. *J. of Personality and Social Psychology*, 77:1296–1312.

J. W. Pennebaker, L. E. Francis, and R. J. Booth, 2001. *LIWC: Linguistic Inquiry and Word Count*.

B. Reeves and C. Nass. 1996. *The Media Equation*. University of Chicago Press.

K. R. Scherer. 1979. Personality markers in speech. In K. R. Scherer and H. Giles, editors, *Social markers in speech*, p. 147–209. Cambridge University Press.

S. Tucker and S. Whittaker. 2004. Accessing multimodal meeting data: Systems, problems and possibilities. *Lecture Notes in Computer Science*, 3361:1–11.

M. Walker and S. Whittaker. 1990. Mixed initiative in dialogue: an investigation into discourse segmentation. In *Proc. of the 28th Annual Meeting of the ACL*, p. 70–78.

Summarizing Speech Without Text Using Hidden Markov Models

Sameer Maskey, Julia Hirschberg
Dept. of Computer Science
Columbia University
New York, NY
{smaskey, julia}@cs.columbia.edu

Abstract

We present a method for summarizing speech documents without using any type of transcript/text in a Hidden Markov Model framework. The hidden variables or states in the model represent whether a sentence is to be included in a summary or not, and the acoustic/prosodic features are the observation vectors. The model predicts the optimal sequence of segments that best summarize the document. We evaluate our method by comparing the predicted summary with one generated by a human summarizer. Our results indicate that we can generate 'good' summaries even when using only acoustic/prosodic information, which points toward the possibility of text-independent summarization for spoken documents.

1 Introduction

The goal of single document text or speech summarization is to identify information from a text or spoken document that summarizes, or conveys the essence of a document. EXTRACTIVE SUMMARIZATION identifies portions of the original document and concatenates these segments to form a summary. How these segments are selected is thus critical to the summarization adequacy.

Many classifier-based methods have been examined for extractive summarization of text and of speech (Maskey and Hirschberg, 2005; Christensen et. al., 2004; Kupiec et. al., 1995). These approaches attempt to classify segments as to whether they should or should not be included in a summary. However, the classifiers used in these methods implicitly assume that the posterior probability for the inclusion of a sentence in the summary is only dependent on the observations for that sentence, and is not affected by previous decisions. Some of these (Kupiec et. al., 1995; Maskey and Hirschberg, 2005) also assume that the features themselves are independent. Such an independence assumption simplifies the training procedure of the models, but it does not appear to model the factors human beings appear to use in generating summaries. In particular, human summarizers seem to take previous decisions into account when deciding if a sentence in the source document should be in the document's summary.

In this paper, we examine a Hidden Markov Model (HMM) approach to the selection of segments to be included in a summary that we believe better models the interaction between extracted segments and their features, for the domain of Broadcast News (BN). In Section 2 we describe related work on the use of HMMs in summarization. We present our own approach in Section 3 and discuss our results in Section 3.1. We conclude in Section 5 and discuss future research.

2 Related Work

Most speech summarization systems (Christensen et. al., 2004; Hori et. al., 2002; Zechner, 2001) use lexical features derived from human or Automatic Speech Recognition (ASR) transcripts as features to select words or sentences to be included in a summary. However, human transcripts are not generally available for spoken documents, and ASR transcripts are errorful. So, lexical features have practical limits as a means of choosing important segments for summarization. Other research efforts have focussed on text-independent approaches to extractive summarization (Ohtake et. al., 2003), which rely upon acoustic/prosodic cues. However, none of these efforts allow for the context-dependence of extractive summarization, such that the inclusion of

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 89–92,
New York, June 2006. ©2006 Association for Computational Linguistics

one word or sentence in a summary depends upon prior selection decisions. While HMMs are used in many language processing tasks, they have not been employed frequently in summarization. A significant exception is the work of Conroy and O'Leary (2001), which employs an HMM model with pivoted QR decomposition for text summarization. However, the structure of their model is constrained by identifying a fixed number of 'lead' sentences to be extracted for a summary. In the work we present below, we introduce a new HMM approach to extractive summarization which addresses some of the deficiencies of work done to date.

3 Using Continuous HMM for Speech Summarization

We define our HMM by the following parameters: $\Omega = 1..N$: The state space, representing a set of states where N is the total number of states in the model; $O = o_{1k}, o_{2k}, o_{3k}, ...o_{Mk}$: The set of observation vectors, where each vector is of size k; $A = \{a_{ij}\}$: The transition probability matrix, where a_{ij} is the probability of transition from state i to state j; $b_j(o_{jk})$: The observation probability density function, estimated by $\Sigma_{k=1}^{M} c_{jk} N(o_{jk}, \mu_{jk}, \Sigma_{jk})$, where o_{jk} denotes the feature vector; $N(o_{jk}, \mu_{jk}, \Sigma_{jk})$ denotes a single Gaussian density function with mean of μ_{jk} and covariance matrix Σ_{jk} for the state j, with M the number of mixture components and c_{jk} the weight of the k^{th} mixture component; $\Pi = \pi_i$: The initial state probability distribution. For convenience, we define the parameters for our HMM by a set λ that represents A, B and Π. We can use the parameter set λ to evaluate $P(O|\lambda)$, i.e. to measure the maximum likelihood performance of the output observables O. In order to evaluate $P(O|\lambda)$, however, we first need to compute the probabilities in the matrices in the parameter set λ

The Markov assumption that state durations have a geometric distribution defined by the probability of self transitions makes it difficult to model durations in an HMM. If we introduce an explicit duration probability to replace self transition probabilities, the Markov assumption no longer holds. Yet, HMMs have been extended by defining state duration distributions called Hidden Semi-Markov Model (HSMM) that has been succesfully used (Tweed et. al., 2005). Similar to (Tweed et. al., 2005)'s use of HSMMs, we want to model the position of a sentence in the source document explicitly. But instead of building an HSMM, we model this positional information by building our position-sensitive HMM in the following way:

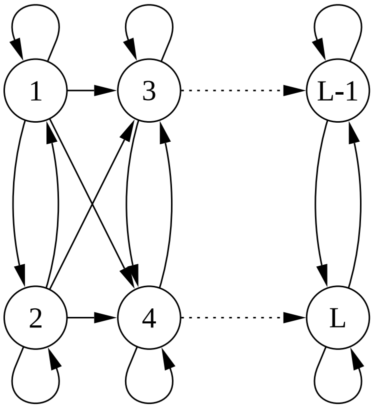


Figure 1: L state position-sensitive HMM

We first discretize the position feature into L number of bins, where the number of sentences in each bin is proportional to the length of the document. We build 2 states for each bin where the second state models the probability of the sentence being included in the document's summary and the other models the exclusion probability. Hence, for L bins we have $2L$ states. For any bin lth where $2l$ and $2l - 1$ are the corresponding states, we remove all transitions from these states to other states except $2(l+1)$ and $2(l+1)-1$. This converts our ergodic L state HMM to an almost Left-to-Right HMM though l states can go back to $l - 1$. This models sentence position in that decisions at the lth state can be arrived at only after decisions at the $(l - 1)th$ state have been made. For example, if we discretize sentence position in document into 10 bins, such that 10% of sentences in the document fall into each bin, then states 13 and 14, corresponding to the seventh bin (.i.e. all positions between 0.6 to 0.7 of the text) can be reached only from states 11, 12, 13 and 14.

The topology of our HMM is shown in Figure 1.

3.1 Features and Training

We trained and tested our model on a portion of the TDT-2 corpus previously used in (Maskey and Hirschberg, 2005). This subset includes 216 stories from 20 CNN shows, comprising 10 hours of audio data and corresponding manual transcript. An annotator generated a summary for each story by extracting sentences. While we thus rely upon human-identified sentence boundaries, automatic sentence detection procedures have been found to perform with reasonable accuracy compared to human performance (Shriberg et. al., 2000).

For these experiments, we extracted only acoustic/prosodic features from the corpus. The intuition behind using acoustic/prosodic features for speech summarization is based on research in speech prosody (Hirschberg, 2002) that humans use acoustic/prosodic variation — expanded pitch range, greater intensity, and timing variation — to indicate the importance of particular segments of their speech. In BN, we note that a change in pitch, amplitude or speaking rate may signal differences in the relative importance of the speech segments produced by anchors and reporters — the professional speakers in our corpus. There is also considerable evidence that topic shift is marked by changes in pitch, intensity, speaking rate and duration of pause (Shriberg et. al., 2000), and new topics or stories in BN are often introduced with content-laden sentences which, in turn, often are included in story summaries.

Our acoustic feature-set consists of 12 features, similar to those used in (Inoue et. al., 2004; Christensen et. al., 2004; Maskey and Hirschberg, 2005). It includes **speaking rate** (the ratio of voiced/total frames); **F0 minimum**, **maximum**, and **mean**; **F0 range** and **slope**; **minimum, maximum**, and **mean RMS energy** (minDB, maxDB, meanDB); **RMS slope** (slopeDB); **sentence duration** (timeLen = endtime - starttime). We extract these features by automatically aligning the annotated manual transcripts with the audio source. We then employ Praat (Boersma, 2001) to extract the features from the audio and produce normalized and raw versions of each. Normalized features were produced by dividing each feature by the average of the feature values for each speaker, where speaker identify was determined from the Dragon speaker segmentation of the TDT-2 corpus. In general, the normalized acoustic features performed better than the raw values.

We used 197 stories from this labeled corpus to train our HMM. We computed the transition probabilities for the matrix A_{NXN} by computing the relative frequency of the transitions made from each state to the other valid states. We had to compute four transition probabilities for each state, i.e. a_{ij} where $j = i, i+1, i+2, i+3$ if i is odd and $j = i-1, i, i+1, i+2$ if i is even. Odd states signify that the sentence should not be included in the summary, while even states signify sentence inclusion. Observation probabilities were estimated using a mixture of Gaussians where the number of mixtures was 12. We computed a $12X1$ matrix for the mean μ and $12X12$ matrices for the covariance matrix Σ for each state. We then computed the maximum likelihood estimates and found the optimal sequence of states to predict the selection of document summaries using the Viterbi algorithm. This approach maximizes the probability of inclusion of sentences at each stage incrementally.

4 Results and Evaluation

We tested our resulting model on a held-out test set of 19 stories. For each sentence in the test set we extracted the 12 acoustic/prosodic features. We built a $12XN$ matrix using these features for N sentences in the story where N was the total length of the story. We then computed the optimal sequence of sentences to include in the summary by decoding our sentence state lattice using the Viterbi algorithm. For all the even states in this sequence we extracted the corresponding segments and concatenated them to produce the summary.

Evaluating summarizers is a difficult problem, since there is great disagreement between humans over what should be included in a summary. Speech summaries are even harder to evaluate because most objective evaluation metrics are based on word overlap. The metric we will use here is the standard information retrieval measure of Precision, Recall and F-measure on sentences. This is a strict metric, since it requires exact matching with sentences in the human summary; we are penalized if we identify sentences similar in meaning but not identical to the gold standard.

We first computed the F-measure of a baseline system which randomly extracts sentences for the summary; this method produces an F-measure of 0.24. To determine whether the positional information captured in our position-sensitive HMM model was useful, we first built a 2-state HMM that models only inclusion/exclusion of sentences from a summary, without modeling sentence position in the document. We trained this HMM on the training corpus described above. We then trained a position-sensitive HMM by first discretizing position into 4 bins, such that each bin includes one-quarter of the sentences in the story. We built an 8-state HMM that captures this positional information. We tested both on our held-out test set. Results are shown in Table 1. Note that recall for the 8-state position-sensitive HMM is 16% better than recall for the 2-state HMM, although precision for the 2-state model is slightly (1%) better than for the 8-state model. The F-measure for the 8-state position-sensitive model represents a slight improvement over the 2-state model, of 1%. These results are encouraging, since, in skewed datasets like documents with their summaries, only a few sentences from a document are usually included in the summary; thus, recall is generally more important than precision in extractive summarization. And, compared to the baseline, the position-sensitive 8-state HMM obtains an F-measure of 0.41, which is 17% higher than the baseline.

ModelType	Precision	Recall	F-Meas
HMM-8state	0.26	0.95	0.41
HMM-2state	0.27	0.79	0.40
Baseline	0.23	0.24	0.24

Table 1: Speech Summarization Results

5 Conclusion

We have shown a novel way of using continuous HMMs for summarizing speech documents without using any lexical information. Our model generates an optimal summary by decoding the state lattice, where states represent whether a sentence should be included in the summary or not. This model is able to take the context and the previous decisions into account generating better summaries. Our results also show that speech can be summarized fairly well using acoustic/prosodic features alone, without lexical features, suggesting that the effect of ASR transcription errors on summarization may be minimized by techniques such as ours.

6 Acknowledgement

We would like to thank Yang Liu, Michel Galley and Fadi Biadsy for helpful comments. This work was funded in part by the DARPA GALE program under a subcontract to SRI International.

References

Boersma P. *Praat, a system for doing phonetics by computer* Glot International 5:9/10, 341-345. 2001.

Christensen H., Kolluru B., Gotoh Y., Renals S. *From text summarisation to style-specific summarisation for broadcast news* Proc. ECIR-2004, 2004

Conroy J. and Leary D.O *Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition* Technical report, University of Maryland, March 2001

Hirschberg J *Communication and Prosody: Functional Aspects of Prosody* Speech Communication, Vol 36, pp 31-43, 2002.

Hori C., Furui S., Malkin R., Yu H., Waibel A.. *Automatic Speech Summarization Applied to English Broadcast News Speech* Proc. of ICASSP 2002, pp. 9-12 .

Inoue A., Mikami T., Yamashita Y. *Improvement of Speech Summarization Using Prosodic Information* Proc. of Speech Prosody 2004, Japan

Kupiec J., Pedersen J.O., Chen F. *A Trainable Document Summarizer* Proc. of SIGIR 1995

Language Data Consortium *“TDT-2 Corpus* Univ. of Pennsylvania.

Maskey S. and Hirschberg J. 2005. *Comparing Lexical, Acoustic/Prosodic, Structural and Discourse Features* Proc. of ICSLP, Lisbon, Portugal.

Ohtake K., Yamamoto K., Toma y., Sado S., Masuyama S. *Newscast Speech Summarization via Sentence Shortening Based on Prosodic Features* Proc. of SSPR pp.167-170. 2003

Shriberg E., Stolcke A., Hakkani-Tur D., Tur G. *Prosody Based Automatic Segmentation of Speech into Sentences and Topics”* Speech Communication 32(1-2) September 2000

Tweed D., Fisher R., Bins J., List T, *Efficient Hidden Semi-Markov Model Inference for Structured Video Sequences* Proc. of (VS-PETS), pp 247-254, Beijing, Oct 2005.

Witbrock M.J. and Mittal V.O. *Ultra-Summarization: A Statistical Approach to Generating Highly Condensed Non-Extractive Summaries* Proc. of SIGIR 1999

Zechner K. *Automatic Generation of Concise Summaries of Spoken Dialogues in Unrestricted Domains* Research and Development in Information Retrieval, 199-207, 2001.

NER Systems that Suit User's Preferences: Adjusting the Recall-Precision Trade-off for Entity Extraction

Einat Minkov, Richard C. Wang
Language Technologies
Institute
Carnegie Mellon University
einat,rcwang@cs.cmu.edu

Anthony Tomasic
Inst. for Software Research
International
Carnegie Mellon University
tomasic@cs.cmu.edu

William W. Cohen
Machine Learning Dept.
Carnegie Mellon University
wcohen@cs.cmu.edu

Abstract

We describe a method based on "tweaking" an existing learned sequential classifier to change the recall-precision tradeoff, guided by a user-provided performance criterion. This method is evaluated on the task of recognizing personal names in email and newswire text, and proves to be both simple and effective.

1 Introduction

Named entity recognition (NER) is the task of identifying named entities in free text—typically personal names, organizations, gene-protein entities, and so on. Recently, sequential learning methods, such as hidden Markov models (HMMs) and conditional random fields (CRFs), have been used successfully for a number of applications, including NER (Sha and Pereira, 2003; Pinto et al., 2003; Mccallum and Lee, 2003). In practice, these methods provide imperfect performance: precision and recall, even for well-studied problems on clean well-written text, reach at most the mid-90's. While performance of NER systems is often evaluated in terms of $F1$ measure (a harmonic mean of precision and recall), this measure may not match user preferences regarding precision and recall. Furthermore, learned NER models may be sub-optimal also in terms of F1, as they are trained to optimize other measures (e.g., loglikelihood of the training data for CRFs).

Obviously, different applications of NER have different requirements for precision and recall. A system might require high precision if it is designed to extract entities as one stage of fact-extraction, where facts are stored directly into a database. On the other hand, a system that generates candidate extractions which are passed to a semi-automatic curation system might prefer higher recall. In some domains, such as anonymization of medical records, high recall is essential.

One way to manipulate an extractor's precision-recall tradeoff is to assign a confidence score to each extracted entity and then apply a global threshold to confidence level. However, confidence thresholding of this sort cannot increase recall. Also, while confidence scores are straightforward to compute in many classification settings, there is no inherent mechanism for computing confidence of a sequential extractor. Culotta and McCallum (2004) suggest several methods for doing this with CRFs.

In this paper, we suggest an alternative simple method for exploring and optimizing the relationship between precision and recall for NER systems. In particular, we describe and evaluate a technique called "extractor tweaking" that optimizes a learned extractor with respect to a specific evaluation metric. In a nutshell, we directly *tweak* the threashold term that is part of any linear classifier, including sequential extractors. Though simple, this approach has not been empirically evaluated before, to our knowledge. Further, although sequential extractors such as HMMs and CRFs are state-of-the-art methods for tasks like NER, there has been little prior research about tuning these extractors' performance to suit user preferences. The suggested algorithm optimizes the system performance per a user-provided

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 93–96,
New York, June 2006.

evaluation criterion, using a linear search procedure. Applying this procedure is not trivial, since the underlying function is not smooth. However, we show that the system's precision-recall rate can indeed be tuned to user preferences given labelled data using this method. Empirical results are presented for a particular NER task—recognizing person names, for three corpora, including email and newswire text.

2 Extractor tweaking

Learning methods such as VP-HMM and CRFs optimize criteria such as margin separation (implicitly maximized by VP-HMMs) or log-likelihood (explicitly maximized by CRFs), which are at best indirectly related to precision and recall. Can such learning methods be modified to more directly reward a user-provided performance metric?

In a non-sequential classifier, a threshold on confidence can be set to alter the precision-recall tradeoff. This is nontrivial to do for VP-HMMs and CRFs. Both learners use dynamic programming to find the label sequence $\mathbf{y} = (y_1, \ldots, y_i, \ldots, y_N)$ for a word sequence $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_N)$ that maximizes the function $\mathbf{W} \cdot \sum_i \mathbf{f}(\mathbf{x}, i, y_{i-1}, y_i)$, where **W** is the learned weight vector and **f** is a vector of features computed from **x**, i, the label y_i for x_i, and the previous label y_{i-1}. Dynamic programming finds the most likely state sequence, and does not output probability for a particular sub-sequence. (Culotta and McCallum, 2004) suggest several ways to generate confidence estimation in this framework. We propose a simpler approach for directly manipulating the learned extractor's precision-recall ratio.

We will assume that the labels y include one label O for "outside any named entity", and let w_0 be the weight for the feature f_0, defined as follows:

$$f_0(\mathbf{x}, i, y_{i-1}, y_i) \equiv \begin{cases} 1 & \text{if } y_i = O \\ 0 & \text{else} \end{cases}$$

If no such feature exists, then we will create one. The NER based on **W** will be sensitive to the value of w_0: large negative values will force the dynamic programming method to label tokens as inside entities, and large positive values will force it to label fewer entities[1].

We thus propose to "tweak" a learned NER by varying the single parameter w_0 systematically so as to optimize some user-provided performance metric. Specifically, we tune w_0 using a a Gauss-Newton line search, where the objective function is iteratively approximated by quadratics.[2] We terminate the search when two adjacent evaluation results are within a 0.01% difference[3].

A variety of performance metrics might be imagined: for instance, one might wish to optimize recall, after applying some sort of penalty for precision below some fixed threshold. In this paper we will experiment with performance metrics based on the (complete) F-measure formula, which combines precision and recall into a single numeric value based on a user-provided parameter β:

$$F(\beta, P, R) = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

A value of $\beta > 1$ assigns higher importance to recall. In particular, F_2 weights recall twice as much as precision. Similarly, $F_{0.5}$ weights precision twice as much as recall.

We consider optimizing both token- and entity-level F_β – awarding partial credit for partially extracted entities and no credit for incorrect entity boundaries, respectively. Performance is optimized over the dataset on which **W** was trained, and tested on a separate set. A key question our evaluation should address is whether the values optimized for the training examples transfer well to unseen test examples, using the suggested approximate procedure.

3 Experiments

3.1 Experimental Settings

We experiment with three datasets, of both email and newswire text. Table 1 gives summary statistics for all datasets. The widely-used *MUC-6* dataset includes news articles drawn from the Wall Street Journal. The *Enron* dataset is a collection of emails extracted from the Enron corpus (Klimt and Yang, 2004), where we use a subcollection of the messages located in folders named "meetings" or "calendar". The *Mgmt-Groups* dataset is a second email

[1] We clarify that w_0 will refer to feature f_0 only, and not to other features that may incorporate label information.

[2] from http://billharlan.com/pub/code/inv.

[3] In the experiments, this is usually within around 10 iterations. Each iteration requires evaluating a "tweaked" extractor on a training set.

collection, extracted from the CSpace email corpus, which contains email messages sent by MBA students taking a management course conducted at Carnegie Mellon University in 1997. This data was split such that its test set contains a different mix of entity names comparing to training exmaples. Further details about these datasets are available elsewhere (Minkov et al., 2005).

	# documents			# names
	Train	Test	# tokens	per doc.
MUC-6	347	30	204,071	6.8
Enron	833	143	204,423	3.0
Mgmt-Groups	631	128	104,662	3.7

Table 1: Summary of the corpora used in the experiments

We used an implementation of Collins' voted-percepton method for discriminatively training HMMs (henceforth, VP-HMM) (Collins, 2002) as well as CRF (Lafferty et al., 2001) to learn a NER. Both VP-HMM and CRF were trained for 20 epochs on every dataset, using a simple set of features such as word identity and capitalization patterns for a window of three words around each word being classified. Each word is classified as either inside or outside a person name.[4]

3.2 Extractor tweaking Results

Figure 1 evaluates the effectiveness of the optimization process used by "extractor tweaking" on the Enron dataset. We optimized models for F_β with different values of β, and also evaluated each optimized model with different F_β metrics. The top graph shows the results for token-level F_β, and the bottom graph shows entity-level F_β behavior. The graph illustates that the optimized model does indeed roughly maximize performance for the target β value: for example, the token-level F_β curve for the model optimized for $\beta = 0.5$ indeed peaks at $\beta = 0.5$ on the test set data. The optimization is only roughly accurate[5] for several possible reasons: first, there are differences between train and test sets; in addition, the line search assumes that the performance metric is smooth and convex, which need not be true. Note that evaluation-metric optimization is less successful for entity-level performance, which behaves less smoothly than token-level performance.

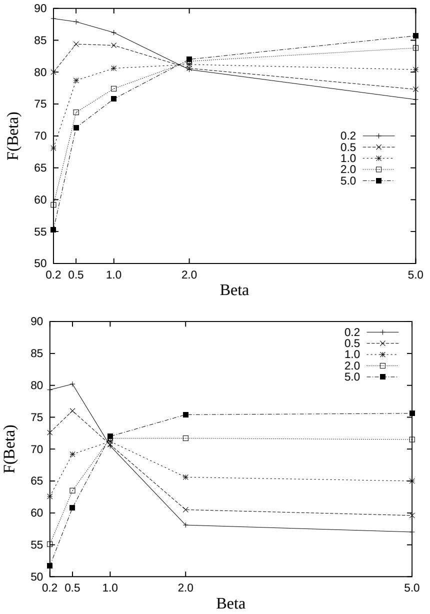


Figure 1: Results of token-level (top) and entity-level (bottom) optimization for varying values of β, for the Enron dataset, VP-HMM. The y-axis gives F in terms of β. β (x-axis) is given in a logarithmic scale.

	Token		Entity	
β	Prec	Recall	Prec	Recall
Baseline	*93.3*	*76.0*	*93.6*	*70.6*
0.2	100	53.2	98.2	57.0
0.5	95.3	71.1	94.4	67.9
1.0	88.6	79.4	89.2	70.9
2.0	81.0	83.9	81.8	70.9
5.0	65.8	91.3	69.4	71.4

Table 2: Sample optimized CRF results, for the MUC-6 dataset and entity-level optimization.

Similar results were obtained optimizing baseline CRF classifiers. Sample results (for MUC-6 only, due to space limitations) are given in Table 2, optimizing a CRF baseline for entity-level F_β. Note that as β increases, recall monotonically increases and precision monotonically falls.

The graphs in Figure 2 present another set of results with a more traditional recall-precision curves. The top three graphs are for token-level F_β optimization, and the bottom three are for entity-level optimization. The solid lines show the token-level and entity-level precision-recall tradeoff obtained by

[4] This problem encoding is basic. However, in the context of this paper we focus on precision-recall trade-off in the general case, avoiding settings' optimization.

[5] E.g, the token-level F_2 curve peaks at $\beta = 5$.

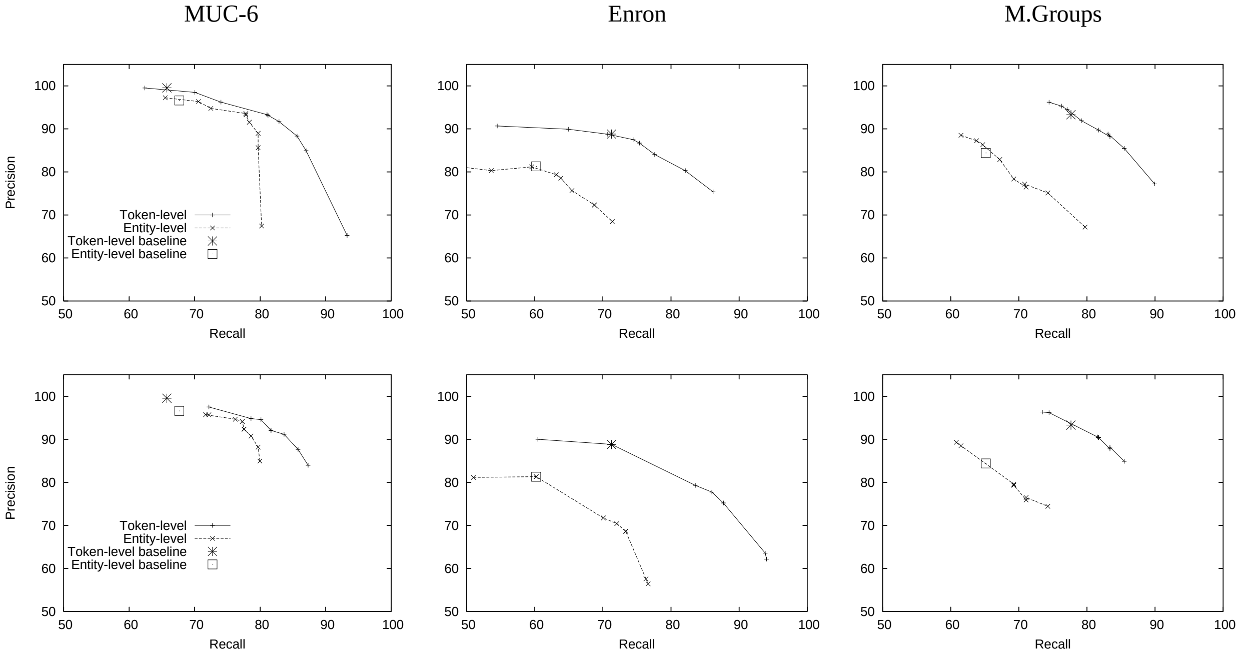


Figure 2: Results for the evaluation-metric model optimization. The top three graphs are for token-level $F(\beta)$ optimization, and the bottom three are for entity-level optimization. Each graph shows the baseline learned VP-HMM and evaluation-metric optimization for different values of β, in terms of both token-level and entity-level performance.

varying[6] β and optimizing the relevant measure for F_β; the points labeled "baseline" show the precision and recall in token and entity level of the baseline model, learned by VP-HMM. These graphs demonstrate that extractor "tweaking" gives approximately smooth precision-recall curves, as desired. Again, we note that the resulting recall-precision tradeoff for entity-level optimization is generally less smooth.

4 Conclusion

We described an approach that is based on modifying an existing learned sequential classifier to change the recall-precision tradeoff, guided by a user-provided performance criterion. This approach not only allows one to explore a recall-precision tradeoff, but actually allows the user to specify a performance metric to optimize, and optimizes a learned NER system for that metric. We showed that using a single free parameter and a Gauss-Newton line search (where the objective is iteratively approximated by quadratics), effectively optimizes two plausible performance measures, token-level F_β and entity-level F_β. This approach is in fact general, as it is applicable for sequential and/or structured learning applications other than NER.

References

M. Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP*.

A. Culotta and A. McCallum. 2004. Confidence estimation for information extraction. In *HLT-NAACL*.

B. Klimt and Y. Yang. 2004. Introducing the Enron corpus. In *CEAS*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

A. Mccallum and W. Lee. 2003. early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *CONLL*.

E. Minkov, R. C. Wang, and W. W. Cohen. 2005. Extracting personal names from emails: Applying named entity recognition to informal text. In *HLT-EMNLP*.

D. Pinto, A. Mccallum, X. Wei, and W. B. Croft. 2003. table extraction using conditional random fields. In *ACM SIGIR*.

F. Sha and F. Pereira. 2003. Shallow parsing with conditional random fields. In *HLT-NAACL*.

[6] We varied β over the values 0.2, 0.5, 0.8, 1, 1.2, 1.5, 2, 3 and 5

Syntactic Kernels for Natural Language Learning: the Semantic Role Labeling Case

Alessandro Moschitti
Department of Computer Science
University of Rome ”Tor Vergata”
Rome, Italy
moschitti@info.uniroma2.it

Abstract

In this paper, we use tree kernels to exploit deep syntactic parsing information for natural language applications. We study the properties of different kernels and we provide algorithms for their computation in linear average time. The experiments with SVMs on the task of predicate argument classification provide empirical data that validates our methods.

1 Introduction

Recently, several tree kernels have been applied to natural language learning, e.g. (Collins and Duffy, 2002; Zelenko et al., 2003; Cumby and Roth, 2003; Culotta and Sorensen, 2004; Moschitti, 2004). Despite their promising results, three general objections against kernel methods are raised: (1) only a subset of the dual space features are relevant, thus, it may be possible to design features in the primal space that produce the same accuracy with a faster computation time; (2) in some cases the high number of features (substructures) of the dual space can produce overfitting with a consequent accuracy decrease (Cumby and Roth, 2003); and (3) the computation time of kernel functions may be too high and prevent their application in real scenarios.

In this paper, we study the impact of the subtree (ST) (Vishwanathan and Smola, 2002), subset tree (SST) (Collins and Duffy, 2002) and partial tree (PT) kernels on Semantic Role Labeling (SRL). The PT kernel is a new function that we have designed to generate larger substructure spaces. Moreover, to solve the computation problems, we propose algorithms which evaluate the above kernels in linear average running time.

We experimented such kernels with Support Vector Machines (SVMs) on the classification of semantic roles of PropBank (Kingsbury and Palmer, 2002) and FrameNet (Fillmore, 1982) data sets. The results show that: (1) the kernel approach provides the same accuracy of the manually designed features. (2) The overfitting problem does not occur although the richer space of PTs does not provide better accuracy than the one based on SST. (3) The average running time of our tree kernel computation is linear.

In the remainder of this paper, Section 2 introduces the different tree kernel spaces. Section 3 describes the kernel functions and our fast algorithms for their evaluation. Section 4 shows the comparative performance in terms of execution time and accuracy.

2 Tree kernel Spaces

We consider three different tree kernel spaces: the subtrees (STs), the subset trees (SSTs) and the novel partial trees (PTs).

An ST of a tree is rooted in any node and includes all its descendants. For example, Figure 1 shows the parse tree of the sentence "Mary brought a cat" together with its 6 STs. An SST is a more general structure since its leaves can be associated with non-terminal symbols. The SSTs satisfy the constraint that grammatical rules cannot be broken. For example, Figure 2 shows 10 SSTs out of 17 of the subtree of Figure 1 rooted in VP. If we relax the non-breaking rule constraint we obtain a more general form of substructures, i.e. the PTs. For example,

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 97–100,
New York, June 2006.

Figure 3 shows 10 out of the total 30 PTs, derived from the same tree as before.

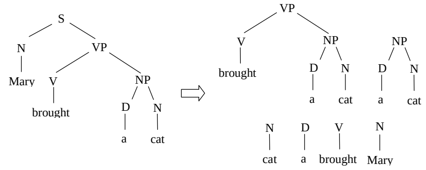


Figure 1: A syntactic parse tree with its subtrees (STs).

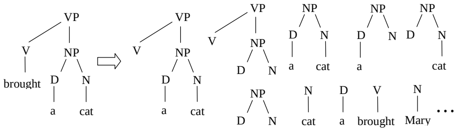


Figure 2: A tree with some of its subset trees (SSTs).

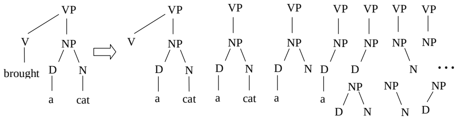


Figure 3: A tree with some of its partial trees (PTs).

3 Fast Tree Kernel Functions

The main idea of tree kernels is to compute the number of common substructures between two trees T_1 and T_2 without explicitly considering the whole fragment space. We designed a general function to compute the ST, SST and PT kernels. Our fast algorithm is inspired by the efficient evaluation of non-continuous subsequences (described in (Shawe-Taylor and Cristianini, 2004)). To further increase the computation speed, we also applied the pre-selection of node pairs which have non-null kernel.

3.1 Generalized Tree Kernel function

Given a tree fragment space $\mathcal{F} = \{f_1, f_2, .., f_{\mathcal{F}}\}$, we use the indicator function $I_i(n)$ which is equal to 1 if the target f_i is rooted at node n and 0 otherwise. We define the general kernel as:

$$K(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2), \qquad (1)$$

where N_{T_1} and N_{T_2} are the sets of nodes in T_1 and T_2, respectively and $\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} I_i(n_1) I_i(n_2)$, i.e. the number of common fragments rooted at the n_1 and n_2 nodes. We can compute it as follows:

- if the node labels of n_1 and n_2 are different then $\Delta(n_1, n_2) = 0$;
- else:

$$\Delta(n_1, n_2) = 1 + \sum_{\vec{J}_1, \vec{J}_2, l(\vec{J}_1) = l(\vec{J}_2)} \prod_{i=1}^{l(\vec{J}_1)} \Delta(c_{n_1}[\vec{J}_{1i}], c_{n_2}[\vec{J}_{2i}]) \qquad (2)$$

where $\vec{J}_1 = \langle J_{11}, J_{12}, J_{13}, ..\rangle$ and $\vec{J}_2 = \langle J_{21}, J_{22}, J_{23}, ..\rangle$ are index sequences associated with the ordered child sequences c_{n_1} of n_1 and c_{n_2} of n_2, respectively, $\vec{J}_{1i}$ and $\vec{J}_{2i}$ point to the i-th children in the two sequences, and $l(\cdot)$ returns the sequence length. We note that (1) Eq. 2 is a convolution kernel according to the definition and the proof given in (Haussler, 1999). (2) Such kernel generates a feature space richer than those defined in (Vishwanathan and Smola, 2002; Collins and Duffy, 2002; Zelenko et al., 2003; Culotta and Sorensen, 2004; Shawe-Taylor and Cristianini, 2004). Additionally, we add the decay factor as follows: $\Delta(n_1, n_2) =$

$$\mu\Big(\lambda^2 + \sum_{\vec{J}_1, \vec{J}_2, l(\vec{J}_1) = l(\vec{J}_2)} \lambda^{d(\vec{J}_1) + d(\vec{J}_2)} \prod_{i=1}^{l(\vec{J}_1)} \Delta(c_{n_1}[\vec{J}_{1i}], c_{n_2}[\vec{J}_{2i}])\Big) \qquad (3)$$

where $d(\vec{J}_1) = \vec{J}_{1l(\vec{J}_1)} - \vec{J}_{11}$ and $d(\vec{J}_2) = \vec{J}_{2l(\vec{J}_2)} - \vec{J}_{21}$. In this way, we penalize subtrees built on child subsequences that contain gaps. Moreover, to have a similarity score between 0 and 1, we also apply the normalization in the kernel space, i.e. $K'(T_1, T_2) = \frac{K(T_1, T_2)}{\sqrt{K(T_1, T_1) \times K(T_2, T_2)}}$. As the summation in Eq. 3 can be distributed with respect to different types of sequences, e.g. those composed by p children, it follows that

$$\Delta(n_1, n_2) = \mu\big(\lambda^2 + \textstyle\sum_{p=1}^{lm} \Delta_p(n_1, n_2)\big), \qquad (4)$$

where Δ_p evaluates the number of common subtrees rooted in subsequences of exactly p children (of n_1 and n_2) and $lm = min\{l(c_{n1}), l(c_{n2})\}$. Note also that if we consider only the contribution of the longest sequence of node pairs that have the same children, we implement the SST kernel. For the STs computation we need also to remove the λ^2 term from Eq. 4.

Given the two child sequences $c_1 a = c_{n_1}$ and $c_2 b = c_{n_2}$ (a and b are the last children), $\Delta_p(c_1 a, c_2 b) =$

$$\Delta(a, b) \times \sum_{i=1}^{|c_1|} \sum_{r=1}^{|c_2|} \lambda^{|c_1| - i + |c_2| - r} \times \Delta_{p-1}(c_1[1:i], c_2[1:r]),$$

where $c_1[1:i]$ and $c_2[1:r]$ are the child subsequences from 1 to i and from 1 to r of c_1 and c_2. If we name the double summation term as D_p, we can rewrite the relation as:

$$\Delta_p(c_1a, c_2b) = \begin{cases} \Delta(a,b)D_p(|c_1|,|c_2|) \text{ if } a = b; \\ 0 \qquad\qquad\qquad\qquad otherwise. \end{cases}$$

Note that D_p satisfies the recursive relation:

$$\begin{aligned} D_p(k,l) \quad &= \Delta_{p-1}(s[1:k], t[1:l]) + \lambda D_p(k, l-1) \\ &+ \lambda D_p(k-1, l) + \lambda^2 D_p(k-1, l-1). \end{aligned}$$

By means of the above relation, we can compute the child subsequences of two sets c_1 and c_2 in $O(p|c_1||c_2|)$. This means that the worst case complexity of the PT kernel is $O(p\rho^2|N_{T_1}||N_{T_2}|)$, where ρ is the maximum branching factor of the two trees. Note that the average ρ in natural language parse trees is very small and the overall complexity can be reduced by avoiding the computation of node pairs with different labels. The next section shows our fast algorithm to find non-null node pairs.

3.2 Fast non-null node pair computation

To compute the kernels defined in the previous section, we sum the Δ function for each pair $\langle n_1, n_2\rangle \in N_{T_1} \times N_{T_2}$ (Eq. 1). When the labels associated with n_1 and n_2 are different, we can avoid evaluating $\Delta(n_1, n_2)$ since it is 0. Thus, we look for a node pair set $N_p = \{\langle n_1, n_2\rangle \in N_{T_1} \times N_{T_2} : label(n_1) = label(n_2)\}$.

To efficiently build N_p, we (i) extract the L_1 and L_2 lists of nodes from T_1 and T_2, (ii) sort them in alphanumeric order and (iii) scan them to find N_p. Step (iii) may require only $O(|N_{T_1}| + |N_{T_2}|)$ time, but, if $label(n_1)$ appears r_1 times in T_1 and $label(n_2)$ is repeated r_2 times in T_2, we need to consider $r_1 \times r_2$ pairs. The formal can be found in (Moschitti, 2006).

4 The Experiments

In these experiments, we study tree kernel performance in terms of average running time and accuracy on the classification of predicate arguments. As shown in (Moschitti, 2004), we can label semantic roles by classifying the smallest subtree that includes the predicate with one of its arguments, i.e. the so called PAF structure.

The experiments were carried out with the SVM-light-TK software available at `http://ai-nlp.info.uniroma2.it/moschitti/` which encodes the fast tree kernels in the SVM-light software (Joachims, 1999). The multiclassifiers were obtained by training an SVM for each class in the ONE-vs.-ALL fashion. In the testing phase, we selected the class associated with the maximum SVM score.

For the ST, SST and PT kernels, we found that the best λ values (see Section 3) on the development set were 1, 0.4 and 0.8, respectively, whereas the best μ was 0.4.

4.1 Kernel running time experiments

To study the FTK running time, we extracted from the Penn Treebank several samples of 500 trees containing exactly n nodes. Each point of Figure 4 shows the average computation time[1] of the kernel function applied to the 250,000 pairs of trees of size n. It clearly appears that the FTK-SST and FTK-PT (i.e. FTK applied to the SST and PT kernels) average running time has linear behavior whereas, as expected, the naïve SST algorithm shows a quadratic curve.

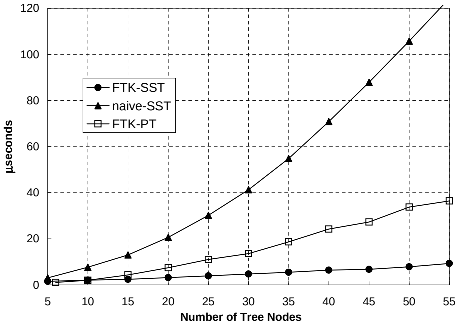


Figure 4: Average time in μseconds for the naïve SST kernel, FTK-SST and FTK-PT evaluations.

4.2 Experiments on SRL dataset

We used two different corpora: PropBank (`www.cis.upenn.edu/∼ace`) along with Penn Treebank 2 (Marcus et al., 1993) and FrameNet. PropBank contains about 53,700 sentences and a fixed split between training and testing used in other researches. In this split, sections from 02 to 21 are used for training, section 23 for testing and section 22 as development set. We considered a total of 122,774 and 7,359 arguments (from *Arg0* to *Arg5*, *ArgA* and *ArgM*) in training and testing, respectively. The tree structures were extracted from the Penn Treebank.

From the FrameNet corpus (`www.icsi.berkeley.edu/∼framenet`) we extracted all

[1] We run the experiments on a Pentium 4, 2GHz, with 1 Gb ram.

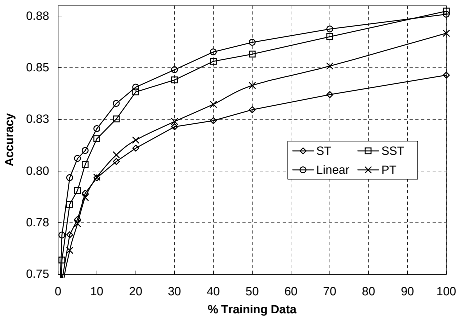


Figure 5: Multiclassifier accuracy according to different training set percentage.

24,558 sentences of the 40 Frames selected for the *Automatic Labeling of Semantic Roles* task of Senseval 3 (`www.senseval.org`). We considered the 18 most frequent roles, for a total of 37,948 examples (30% of the sentences for testing and 70% for training/validation). The sentences were processed with the Collins' parser (Collins, 1997) to generate automatic parse trees.

We run ST, SST and PT kernels along with the linear kernel of standard features (Carreras and Màrquez, 2005) on PropBank training sets of different size. Figure 5 illustrates the learning curves associated with the above kernels for the SVM multiclassifiers.

The tables 1 and 2 report the results, using all available training data, on PropBank and FrameNet test sets, respectively. We note that: (1) the accuracy of PTs is almost equal to the one produced by SSTs as the PT space is a hyperset of SSTs. The small difference is due to the poor relevance of the substructures in the PT $-$ SST set, which degrade the PT space. (2) The high F_1 measures of tree kernels on FrameNet suggest that they are robust with respect to automatic parse trees.

Moreover, the learning time of SVMs using FTK for the classification of one large argument (Arg 0) is much lower than the one required by naïve algorithm. With all the training data FTK terminated in 6 hours whereas the naïve approach required more than 1 week. However, the *complexity burden* of working in the dual space can be alleviated with recent approaches proposed in (Kudo and Matsumoto, 2003; Suzuki et al., 2004).

Finally, we carried out some experiments with the combination between linear and tree kernels and we found that tree kernels improve the models based on manually designed features by 2/3 percent points, thus they can be seen as a useful tactic to boost system accuracy.

Args	Linear	ST	SST	PT
Acc.	87.6	84.6	87.7	86.7

Table 1: Evaluation of kernels on PropBank data and gold parse trees.

Roles	Linear	ST	SST	PT
Acc.	82.3	80.0	81.2	79.9

Table 2: Evaluation of kernels on FrameNet data encoded in automatic parse trees.

References

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of CoNLL05*.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL02*.

Michael Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of the ACL97*.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL04*.

Chad Cumby and Dan Roth. 2003. Kernel methods for relational learning. In *Proceedings of ICML03*.

Charles J. Fillmore. 1982. Frame semantics. In *Linguistics in the Morning Calm*.

D. Haussler. 1999. Convolution kernels on discrete structures. Technical report ucs-crl-99-10, University of California Santa Cruz.

T. Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*.

Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. In *Proceedings of LREC02*.

Taku Kudo and Yuji Matsumoto. 2003. Fast methods for kernel-based text analysis. In *Proceedings of ACL03*.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *proceedings of ACL04*.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proceedings of EACL06*.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Jun Suzuki, Hideki Isozaki, and Eisaku Maeda. 2004. Convolution kernels with feature selection for natural language processing tasks. In *Proceedings of ACL04*.

S.V.N. Vishwanathan and A.J. Smola. 2002. Fast kernels on strings and trees. In *Proceedings of NIPS02*.

D. Zelenko, C. Aone, and A. Richardella. 2003. Kernel methods for relation extraction. *JMLR*.

Accurate Parsing of the Proposition Bank

Gabriele Musillo
Depts of Linguistics and Computer Science
University of Geneva
2 Rue de Candolle
1211 Geneva 4 Switzerland
musillo4@etu.unige.ch

Paola Merlo
Department of Linguistics
University of Geneva
2 Rue de Candolle
1211 Geneva 4 Switzerland
merlo@lettres.unige.ch

Abstract

We integrate PropBank semantic role labels to an existing statistical parsing model producing richer output. We show conclusive results on joint learning and inference of syntactic and semantic representations.

1 Introduction

Recent successes in statistical syntactic parsing based on supervised techniques trained on a large corpus of syntactic trees (Collins, 1999; Charniak, 2000; Henderson, 2003) have brought the hope that the same approach could be applied to the more ambitious goal of recovering the propositional content and the frame semantics of a sentence. Moving towards a shallow semantic level of representation has immediate applications in question-answering and information extraction. For example, an automatic flight reservation system processing the sentence *I want to book a flight from Geneva to New York* will need to know that *from Geneva* indicates the origin of the flight and *to New York* the destination.

(Gildea and Jurafsky, 2002) define this shallow semantic task as a classification problem where the semantic role to be assigned to each constituent is inferred on the basis of probability distributions of syntactic features extracted from parse trees. They use learning features such as phrase type, position, voice, and parse tree path. Consider, for example, a sentence such as *The authority dropped at midnight Tuesday to $ 2.80 trillion* (taken from section 00 of PropBank (Palmer et al., 2005)). The fact that *to $ 2.80 trillion* receives a direction semantic label is highly correlated to the fact that it is a Prepositional Phrase (PP), that it follows the verb *dropped*, a verb of change of state requiring an end point, that the verb is in the active voice, and that the PP is in a certain tree configuration with the governing verb. All the recent systems proposed for semantic role labelling (SRL) follow this same assumption (CoNLL, 2005).

The assumption that syntactic distributions will be predictive of semantic role assignments is based on linking theory. Linking theory assumes the existence of a hierarchy of semantic roles which are mapped by default on a hierarchy of syntactic positions. It also shows that regular mappings from the semantic to the syntactic level can be posited even for those verbs whose arguments can take several syntactic positions, such as psychological verbs, locatives, or datives, requiring a more complex theory. (See (Hale and Keyser, 1993; Levin and Rappaport Hovav, 1995) among many others.) If the internal semantics of a predicate determines the syntactic expressions of constituents bearing a semantic role, it is then reasonable to expect that knowledge about semantic roles in a sentence will be informative of its syntactic structure, and that learning semantic role labels at the same time as parsing will be beneficial to parsing accuracy.

We present work to test the hypothesis that a current statistical parser (Henderson, 2003) can output rich information comprising both a parse tree and semantic role labels robustly, that is without any significant degradation of the parser's accuracy on the original parsing task. We achieve promising results both on the simple parsing task, where the accuracy of the parser is measured on the standard Parseval measures, and also on the parsing task where more

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 101–104,
New York, June 2006.

complex labels comprising both syntactic labels and semantic roles are taken into account.

These results have several consequences. First, we show that it is possible to build a single integrated system successfully. This is a meaningful achievement, as a task combining semantic role labelling and parsing is more complex than simple syntactic parsing. While the shallow semantics of a constituent and its structural position are often correlated, they sometimes diverge. For example, some nominal temporal modifiers occupy an object position without being objects, like *Tuesday* in the Penn Treebank representation of the sentence above. The indirectness of the relation is also confirmed by the difficulty in exploiting semantic information for parsing. Previous attempts have not been successful. (Klein and Manning, 2003) report a reduction in parsing accuracy of an unlexicalised PCFG from 77.8% to 72.9% in using Penn Treebank function labels in training. The two existing systems that use function labels sucessfully, either inherit Collins' modelling of the notion of complement (Gabbard, Kulick and Marcus, 2006) or model function labels directly (Musillo and Merlo, 2005). Furthermore, our results indicate that the proposed models are robust. To model our task accurately, additional parameters must be estimated. However, given the current limited availability of annotated treebanks, this more complex task will have to be solved with the same overall amount of data, aggravating the difficulty of estimating the model's parameters due to sparse data.

2 The Data and the Extended Parser

In this section we describe the augmentations to our base parsing models necessary to tackle the joint learning of parse tree and semantic role labels.

PropBank encodes propositional information by adding a layer of argument structure annotation to the syntactic structures of the Penn Treebank (Marcus et al., 1993). Verbal predicates in the Penn Treebank (PTB) receive a label REL and their arguments are annotated with abstract semantic role labels A0-A5 or AA for those complements of the predicative verb that are considered arguments while those complements of the verb labelled with a semantic functional label in the original PTB receive the composite semantic role label AM-X, where X stands for labels such as LOC, TMP or ADV, for locative, temporal and adverbial modifiers respectively. PropBank uses two levels of granularity in its annotation, at least conceptually. Arguments receiving labels A0-A5 or AA do not express consistent semantic roles and are specific to a verb, while arguments receiving an AM-X label are supposed to be adjuncts, and the roles they express are consistent across all verbs.

To achieve the complex task of assigning semantic role labels while parsing, we use a family of state-of-the-art history-based statistical parsers, the Simple Synchrony Network (SSN) parsers (Henderson, 2003), which use a form of left-corner parse strategy to map parse trees to sequences of derivation steps. These parsers do not impose any a priori independence assumptions, but instead smooth their parameters by means of the novel SSN neural network architecture. This architecture is capable of inducing a finite history representation of an unbounded sequence of derivation steps, which we denote $h(d_1, \ldots, d_{i-1})$. The representation $h(d_1, \ldots, d_{i-1})$ is computed from a set f of handcrafted features of the derivation move d_{i-1}, and from a finite set D of recent history representations $h(d_1, \ldots, d_j)$, where $j < i - 1$. Because the history representation computed for the move $i - 1$ is included in the inputs to the computation of the representation for the next move i, virtually any information about the derivation history could flow from history representation to history representation and be used to estimate the probability of a derivation move. In our experiments, the set D of earlier history representations is modified to yield a model that is sensitive to regularities in structurally defined sequences of nodes bearing semantic role labels, within and across constituents. For more information on this technique to capture structural domains, see (Musillo and Merlo, 2005) where the technique was applied to function parsing. Given the hidden history representation $h(d_1, \cdots, d_{i-1})$ of a derivation, a normalized exponential output function is computed by the SSNs to estimate a probability distribution over the possible next derivation moves d_i.

To exploit the intuition that semantic role labels are predictive of syntactic structure, we must provide semantic role information as early as possible to the parser. Extending a technique presented in (Klein and Manning, 2003) and adopted in (Merlo and Musillo, 2005) for function labels with state-of-the-art results, we split some part-of-speech tags into tags marked with AM-X semantic role labels. As a result, 240 new POS tags were introduced to partition the original tag set which consisted of 45 tags. Our augmented model has a total of 613 non-terminals to represent both the PTB and PropBank labels, instead of the 33 of the original SSN parser. The 580 newly introduced labels consist of a standard PTB label followed by one or more PropBank semantic roles, such as PP-AM-TMP or NP-A0-A1. These augmented tags and the new non-terminals are included in the set f, and will influence bottom-up projection of structure directly.

These newly introduced fine-grained labels fragment our PropBank data. To alleviate this problem, we enlarge the set f with two additional binary features. One feature decides whether a given preterminal or nonterminal label is a semantic role label belonging to the set comprising the labels A0-A5 and AA. The other feature indicates if a given label is a semantic role label of type AM-X, or otherwise. These features allow the SSN to generalise in several ways. All the constituents bearing an A0-A5 and AA labels will have a common feature. The same will be true for all nodes bearing an AM-X label. Thus, the SSN can generalise across these two types of labels. Finally, all constituents that do not bear any label will now constitute a class, the class of the nodes for which these two features are false.

3 Experiments and Discussion

Our extended semantic role SSN parser was trained on sections 2-21 and validated on section 24 from the PropBank. Testing data are section 23 from the CoNLL-2005 shared task (Carreras and Marquez, 2005).

We perform two different evaluations on our model trained on PropBank data. We distinguish between two parsing tasks: the PropBank parsing task and the PTB parsing task. To evaluate the former parsing task, we compute the standard Parseval measures of labelled recall and precision of constituents, taking into account not only the 33 original labels, but also the newly introduced PropBank labels. This evaluation gives us an indication of how accurately and exhaustively we can recover this richer set of non-terminal labels. The results, computed on the testing data set from the PropBank, are shown in the PropBank column of Table 1, first line. To evaluate the PTB task, we ignore the set of PropBank semantic role labels that our model assigns to constituents (PTB column of Table 1, first line to be compared to the third line of the same column).

To our knowledge, no results have yet been published on parsing the PropBank.[1] Accordingly, it is not possible to draw a straightforward quantitative comparison between our PropBank SSN parser and other PropBank parsers. However, state-of-the-art semantic role labelling systems (CoNLL, 2005) use parse trees output by state-of-the-art parsers (Collins, 1999; Charniak, 2000), both for training and testing, and return partial trees annotated with semantic role labels. An indirect way of comparing our parser with semantic role labellers suggests itself. [2] We merge the partial trees output by a semantic role labeller with the output of the parser on which it was trained, and compute PropBank parsing performance measures on the resulting parse trees. The third line, PropBank column of Table 1 reports such measures summarised for the five best semantic role labelling systems (Punyakanok et al., 2005b; Haghighi et al., 2005; Pradhan et al., 2005; Marquez et al., 2005; Surdeanu and Turmo, 2005) in the CoNLL 2005 shared task. These systems all use (Charniak, 2000)'s parse trees both for training and testing, as well as various other information sources including sets of n-best parse trees, chunks, or named entities. Thus, the partial trees output by these systems were merged with the parse trees returned by Charniak's parser (second line, PropBank column).[3]

These results jointly confirm our initial hypothe-

[1] (Shen and Joshi, 2005) use PropBank labels to extract LTAG spinal trees to train an incremental LTAG parser, but they do not parse PropBank. Their results on the PTB are not directly comparable to ours as calculated on dependecy relations and obtained using gold POS.

[2] Current work aims at extending our parser to recovering the argument structure for each verb, supporting a direct comparison to semantic role labellers.

[3] Because of differences in tokenisations, we retain only 2280 sentences out of the original 2416.

	PTB	PropBank
SSN+Roles model	89.0	82.8
CoNLL five best	-	83.3–84.1
Henderson 03 SSN	89.1	-

Table 1: Percentage F-measure of our SSN parser on PTB and PropBank parsing, compared to the original SSN parser and to the best CoNLL 2005 SR labellers.

sis. The performance on the parsing task (PTB column) does not appreciably deteriorate compared to a current state-of-the-art parser, even if our learner can output a much richer set of labels, and therefore solves a considerably more complex problem, suggesting that the relationship between syntactic PTB parsing and semantic PropBank parsing is strict enough that an integrated approach to the problem of semantic role labelling is beneficial. Moreover, the results indicate that we can perform the more complex PropBank parsing task at levels of accuracy comparable to those achieved by the best semantic role labellers (PropBank column). This indicates that the model is robust, as it has been extended to a richer set of labels successfully, without increase in training data. In fact, the limited availability of data is increased further by the high variability of the argumental labels A0-A5 whose semantics is specific to a given verb or a given verb sense.

Methodologically, these initial results on a joint solution to parsing and semantic role labelling provide the first direct test of whether parsing is necessary for semantic role labelling (Gildea and Palmer, 2002; Punyakanok et al., 2005a). Comparing semantic role labelling based on chunked input to the better semantic role labels retrieved based on parsed trees, (Gildea and Palmer, 2002) conclude that parsing is necessary. In an extensive experimental investigation of the different learning stages usually involved in semantic role labelling, (Punyakanok et al., 2005a) find instead that sophisticated chunking can achieve state-of-the-art results. Neither of these pieces of work actually used a parser to do SRL. Their investigation was therefore limited to establishing the usefulness of syntactic features for the SRL task. Our results do not yet indicate that parsing is beneficial to SRL, but they show that the joint task can be performed successfully.

Acknowledgements We thank the Swiss NSF for supporting this research under grant number 101411-105286/1, James Henderson and Ivan Titov for sharing their SSN software, and Xavier Carreras for providing the CoNLL-2005 data.

References

X. Carreras and L. Marquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. *Procs of CoNLL-2005*.

E. Charniak. 2000. A maximum-entropy-inspired parser. *Procs of NAACL'00*, pages 132–139, Seattle, WA.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, Pennsylvania.

CoNLL. 2005. *Ninth Conference on Computational Natural Language Learning* (CoNLL-2005), Ann Arbor, MI.

R. Gabbard, S. Kulick and M. Marcus 2006. Fully parsing the Penn Treebank. *Procs of NAACL'06*, New York, NY.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

D. Gildea and M. Palmer. 2002. The necessity of parsing for predicate argument recognition. *Procs of ACL 2002*, 239–246, Philadelphia, PA.

A. Haghighi, K. Toutanova, and C. Manning. 2005. A joint model for semantic role labeling. *Procs of CoNLL-2005*, Ann Arbor, MI.

K. Hale and J. Keyser. 1993. On argument structure and the lexical representation of syntactic relations. In K. Hale and J. Keyser, editors, *The View from Building 20*, 53–110. MIT Press.

J. Henderson. 2003. Inducing history representations for broad-coverage statistical parsing. *Procs of NAACL-HLT'03*, 103–110, Edmonton, Canada.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. *Procs of ACL'03*, 423–430, Sapporo, Japan.

B. Levin and M. Rappaport Hovav. 1995. *Unaccusativity*. MIT Press, Cambridge, MA.

M. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

L. Marquez, P. Comas, J. Gimenez, and N. Catala. 2005. Semantic role labeling as sequential tagging. *Procs of CoNLL-2005*.

P. Merlo and G. Musillo. 2005. Accurate function parsing. *Procs of HLT/EMNLP 2005*, 620–627, Vancouver, Canada.

G.Musillo and P. Merlo. 2005. Lexical and structural biases for function parsing. *Procs of IWPT'05*, 83–92, Vancouver, Canada.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31:71–105.

S. Pradhan, K. Hacioglu, W. Ward, J. Martin, and D. Jurafsky. 2005. Semantic role chunking combining complementary syntactic views. *Procs of CoNLL-2005*.

V. Punyakanok, D. Roth, and W. Yih. 2005a. The necessity of syntactic parsing for semantic role labeling. *Procs of IJCAI'05*, Edinburgh, UK.

V. Punyakanok, P. Koomen, D. Roth, and W. Yih. 2005b. Generalized inference with multiple semantic role labeling systems. *Procs of CoNLL-2005*.

L.Shen and A. Joshi. 2005. Incremental LTAG parsing. *Procs of HLT/EMNLP 2005*, Vancouver, Canada.

M. Surdeanu and J. Turmo. 2005. Semantic role labeling using complete syntactic analysis. *Procs of CoNLL-2005*.

Using Semantic Authoring for Blissymbols Communication Boards

Yael Netzer

Dept. of Computer Science
Ben Gurion University
Beer Sheva, Israel
yaeln@cs.bgu.ac.il

Michael Elhadad

Dept. of Computer Science
Ben Gurion University
Beer Sheva, Israel
elhadad@cs.bgu.ac.il

Abstract

Natural language generation (NLG) refers to the process of producing text in a spoken language, starting from an internal knowledge representation structure. Augmentative and Alternative Communication (AAC) deals with the development of devices and tools to enable basic conversation for language-impaired people. We present an applied prototype of an AAC-NLG system generating written output in English and Hebrew from a sequence of Bliss symbols. The system does not "translate" the symbols sequence, but instead, it dynamically changes the communication board as the choice of symbols proceeds according to the syntactic and semantic content of selected symbols, generating utterances in natural language through a process of semantic authoring.

1 Introduction

People who suffer from severe language impairments lack the ability to express themselves through natural usage of language and cannot achieve various forms of communication. The field of Augmentative and Alternative Communication (AAC) is concerned with methods that can be added to the natural communication. In the most common form, iconic symbols are presented on a display (or a communication board). Communication is conducted by the sequential selection of symbols on the display (with vocal output when available), which are then interpreted by the partner in the interaction.

AAC devices are characterized by three aspects: (i) Selection method *i.e.*, the physical choice of symbols on the communication board; (ii) input language and (iii) output medium. In a computerized system, as (McCoy and Hershberger, 1999) mention, a processing method aspect is added to this list. This method refers to the process which creates the output once symbols are inserted.

We specifically study the set of symbols (as an input language) called *Blissymbolics* (*Bliss* in short). *Bliss* is a graphic meaning-referenced language, created by Charles Bliss to be used as a written universal language (Bliss, 1965); since 1971, *Blissymbols* are used for communication with severely language-impaired children. Bliss is designed to be a written-only language, with non-arbitrary symbols. Symbols are constructed from a composition of atomic icons. Because words are structured from semantic components, the graphic representation by itself provides information on words' connectivity [1].

In the last decade, several systems that integrate NLG techniques for AAC systems have been developed ((McCoy, 1997), (Vaillant, 1997) for example). These systems share a common architecture: a telegraphic input sequence (words or symbols) is first parsed, and then a grammatical sentence that represents the message is generated.

This paper presents an NLG-AAC system that generates messages through a controlled process of authoring, where each step in the selection of symbols is controlled by the input specification defined

[1] See http://www.bci.org for reference on the language

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 105–108,
New York, June 2006.

for the linguistic realizer.

2 Generating Messages via Translation

A major difficulty when parsing a telegraphic sequence of words or symbols, is that many of the hints that are used to capture the structure of the text and, accordingly, the meaning of the utterance, are missing. Moreover, as an AAC device is usually used for real-time conversation, the interpretation of utterances relies heavily on pragmatics – time of mentioned events, reference to the immediate environment.

Previous works dealing with translating telegraphic text, such as (Grishman and Sterling, 1989), (Lee et al., 1997) requires to identify dependency relations among the tokens of the telegraphic input. Rich lexical knowledge is needed to identify possible dependencies in a given utterance, *i.e.*, to find the predicate and to apply constraints, such as selectional restrictions to recognize its arguments.

Similar methods were used for AAC applications, COMPANSION (McCoy, 1997) for example – where the telegraphic text is expanded to full sentences, using a *word order parser,* and *a semantic parser* to build the case frame structure of the verb in the utterance, filling the slots with the rest of the content words given. The system uses the semantic representation to re-generate fluent text, relying on lexical resources and NLG techniques.

The main questions at stake in this approach are how good can a semantic parser be, in order to reconstruct the full structure of the sentence from telegraphic input and are pragmatic gaps in the given telegraphic utterances recoverable in general.

3 Generating Messages via Semantic Authoring

Our approach differs from previous NLG-AAC systems in that, with the model of semantic authoring (Biller et al., 2005), we intervene during the *process* of composing the input sequence, and thus can provide early feedback (in the form of display composition and partial text feedback), while preventing the need for parsing a telegraphic sequence.

Semantic parsing is avoided by constructing a semantic structure explicitly while the user inputs the sequence incrementally. It combines three aspects into an integrated approach for the design of an AAC system:

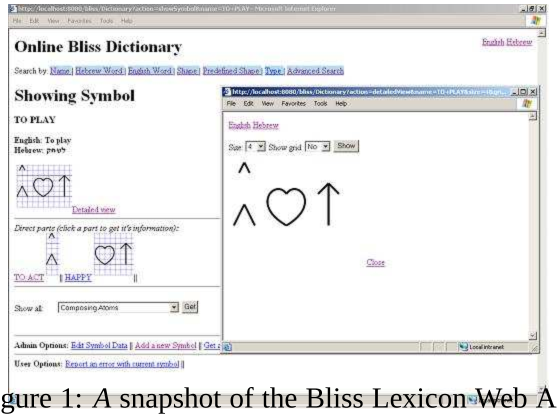
- Semantic authoring drives a natural language realization system and provides rich semantic input.
- A display is updated on the fly as the authoring system requires the user to select options.
- Ready-made inputs, corresponding to predefined pragmatic contexts are made available to the user as semantic templates.

In this method, each step of input insertion is controlled by a set of constraints and rules, which are drawn from an ontology. The system offers, at each step, only possible complements to a small set of concepts. For example, if the previous symbol denotes a verb which requires an instrumental theme, only symbols that can function as instruments are presented on the current display. Other symbols are accessible through navigation operations, which are interpreted in the context of the current partial semantic specification. The general context of each utterance or conversation can be determined by the user, therefore narrowing the number of symbols displayed in the board.

The underlying process of message generation is based on layered lexical knowledge bases (LKB) and an ontology. The ontology serves as a basis for the semantic authoring process; it includes a hierarchy of concepts and relations, and the information it encodes interacts with the conceptual graphs processing performed as part of content determination and lexical choice. The ontology was acquired with a semi-automatic tool, which relies on WordNet (Miller, 1995) and VerbNet (Kipper et al., 2000).

We designed and implemented the **Bliss lexicon** for both Hebrew and English. The lexicon can be used either as a stand-alone lexicon or as part of an application through an API. The design of the lexicon takes advantage of the unique properties of the language. The Bliss lexicon provides the list of symbols accessible to the user, along with their graphic representation, semantic information, and the mapping of symbols to English and Hebrew words. The lexicon can be searched by keyword (*learn*), or by semantic/graphic component: searching all words in the lexicon that contain both *food* and *meat* returns the symbols *hamburger, hot-dog, meatball etc.* (see

Fig. 1). The lexicon currently includes 2,200 entries.



Online Bliss Dictionary
English Hebrew
Search by: Name | Hebrew Word | English Word | Shape | Predefined Shape | Type | Advanced Search
Showing Symbol
File Edit View Favorites Tools Help
Detailed view
Direct parts (click a part to get it's information):
TO ACT || HAPPY ||
Close
Show all: Composing Atoms Get
Size: 4 Show grid No Show
Admin Options: Edit Symbol Data || Add a new Symbol || Get a
User Options: Report an error with current symbol ||
Local intranet

TO PLAY

English: To play
Hebrew: לשחק

Figure 1: *A* snapshot of the Bliss Lexicon Web Application

The core of the processing machinery of the AAC message generation system is based on **SAUT** (Biller et al., 2005) – an authoring system for logical forms encoded as conceptual graphs (CG). The system belongs to the family of WYSIWYM (What You See Is What You Mean) (Power and Scott, 1998) text generation systems: logical forms are entered interactively and the corresponding linguistic realization of the expressions is generated in several languages. The system maintains a model of the discourse context corresponding to the authored documents to enable reference planning in the generation process.

Generating language from pictorial inputs, and specifically from Bliss symbols using semantic authoring in the WYSIWYM approach is not only a pictorial application of the textual version, but it also addresses specific needs of augmentative communication.

As was mentioned above, generating text from a telegraphic message for AAC usage must take the context of the conversation into account. We address this problem in two manners: (1) adding pre-defined inputs into the system (yet allowing accurate text generation that considers syntactic variations), and (2) enabling the assignment of default values to each conversation (such as participants, tense, mood). We also take advantage of the unique properties of the Bliss symbols; the set of symbols that are offered in each display can be filtered using their semantic/graphical connectivity; the reduction of the number of possible choices that are to be made by the user in each step of the message generation affects the cognitive load and can affect the rate of communication.

4 Evaluation

We evaluate our system as an AAC application for message generation from communication boards. From an NLG evaluation perspective, this corresponds to an intrinsic evaluation, *i.e.* judging quality criteria of the generated text and its adequacy relative to the input (Bangalore et al., 1998). Since the prototype of our system is not yet adjusted to interact with alternative pointing devices, we could not test it on actual Bliss users, and could not perform a full extrinsic (task-based) evaluation.

However, as argued in (Higginbotham, 1995), evaluations of AAC systems with nondisabled subjects, when appropriately used, is easier to perform, and in some cases provide superior results. Higginbotham's claims rely on the observation that the methods of message production are not unique to AAC users and analogous communication situations exist both for disabled and nondisabled users. Nondisabled subjects can contribute to the understanding of the cognitive processes underlying the acquisition of symbol and device performance competencies. We believe that the evaluation of efficiency for non-AAC users should be served as baseline.

The approach we offer for message generation requires users to plan their sentences abstractly. (McCoy and Hershberger, 1999) points that novel systems may be found to slow communication but to increase literacy skills. We therefore tested both speed of message generation and semantic coverage (the capability to generate a given message correctly).

The usage of semantic authoring was evaluated on nondisabled subjects through a user study of 10 subjects. This provides a reliable approximation of the learning curve and usability of the system in general (Biller et al., 2005).

In order to evaluate the keystroke savings of the system we have collected a set of 19 sentences written in Bliss and their full English correspondents. We compared the number of the words in the English sentences with the number of choices needed to generate the sentence with our system. The total number of choice steps is 133, while the total number of words in the sentences is 122. This simple ratio shows no improvement of keystrokes saving using our system. Savings, therefore, must be calculated in terms of narrowing the choice possibilities in each step of the process.

However, counting the number of words does not include morphology which in Bliss symbols requires additional choices. We have counted the words in the sentences considering morphology markers of inflections as additional words, all summing to 138, as was suggested in (McCoy and Hershberger, 1999).

Assuming a display with 50 symbols (and additional keys for functions) – a vocabulary of requires 50 different screens. Assuming symbols are organized by frequencies (first screens present the most frequently used words) or by semantic domain.

The overall number of selections is reduced using our communication board since the selectional restrictions narrow the number of possible choices that can be made at each step. The extent to which selection time can be reduced at each step depends on the application domain and the ontology structure. We cannot evaluate it in general, but expect that a well-structured ontology could support efficient selection mechanisms, by grouping semantically related symbols in dedicated displays.

In addition, the semantic authoring approach can generate fluent output in other languages (English and Hebrew, beyond the Bliss sequence – without requiring noisy translation). We also hypothesize that ontologically motivated grouping of symbols could speed up each selection step – but this claim must be assessed empirically in a task-based extrinsic evaluation, which remains to be done in the future.

We are now building the environment for AAC users with cooperation with ISAAC-ISRAEL [2], in order to make the system fully accessible and to be tested by AAC-users. However, this work is still in progress. Once this will be achieved, full evaluation of the system will be plausible.

5 Conclusions and Future Work

This work offers a new approach for message generation in the context of AAC displays using semantic authoring and preventing the need to parse and re-generate. We have designed and implemented a Bliss lexicon for both Hebrew and English, which can either be used a stand-alone lexicon for reference usage or as a part of an application.

Future work includes an implementation of a system with full access for alternative devices, expansion of the underlying lexicon for Hebrew generation, and adding voice output.

[2] Israeli chapter of the International Society for Augmentative and Alternative Communication

References

Srinivas Bangalore, Anoop Sarkar, Christy Doran, and Beth-Ann Hockey. 1998. Grammar and parser evaluation in the XTAG project. In *Proc. of Workshop on Evaluation of Parsing Systems*, Granada, Spain, May.

Ofer Biller, Michael Elhadad, and Yael Netzer. 2005. Interactive authoring of logical forms for multilingual generation. In *Proc. of the 10th workshop of ENLG*, Aberdeen, Scotland.

Charles K. Bliss. 1965. *Semantography (Blissymbolics)*. Semantography Press, Sidney.

Ralph Grishman and John Sterling. 1989. Analyzing telegraphic messages. In *Proc. of DARPA Speech and Natural Language Workshop*, pages 204–208, Philadelphia, Februrary.

D. Jeffery Higginbotham. 1995. Use of nondisabled subjects in AAC research: Confessions of a research infidel. *AAC Augmentative and Alternative Communication*, 11, March. AAC Research forum.

K. Kipper, H. Trang Dang, and M. Palmer. 2000. Class-based construction of a verb lexicon. In *Proceeding of AAAI-2000*.

Young-Suk Lee, Clifford Weinstein, Stephanie Seneff, and Dinesh Tummala. 1997. Ambiguity resolution for machine translation of telegraphic messages. In *Proc. of the 8th conference on EACL*, pages 120–127.

Kathleen F. McCoy and Dave Hershberger. 1999. The role of evaluation in bringing NLP to AAC: A case to consider. In Filip T. Loncke, John Clibbens, Helen H. Arvidson, and Lyle L. Lloyd, editors, *AAC: New Directions in Research and Practice*, pages 105–122. Whurr Publishers, London.

Kathleen F. McCoy. 1997. Simple NLP techiques for expanding telegraphic sentences. In *Proc. of workshop on NLP for Communication Aids*, Madrid, July. ACL/EACL.

George A. Miller. 1995. WORDNET: a lexical database for English. *Commun. ACM*, 38(11):39–41.

Roger Power and Donia Scott. 1998. Multilingual authoring using feedback texts. In *Proc. of COLING-ACL 98*, Montreal, Canada.

Pascal Vaillant. 1997. A semantic-based communication system for dysphasic subjects. In *Proc. of the 6th conference on AI in Medicine Europe (AIME'97)*, Grenoble, France, March.

Extracting Salient Keywords from Instructional Videos Using Joint Text, Audio and Visual Cues

Youngja Park and Ying Li
IBM T.J. Watson Research Center
Hawthorne, NY 10532
{young_park, yingli}@us.ibm.com

Abstract

This paper presents a multi-modal feature-based system for extracting salient keywords from transcripts of instructional videos. Specifically, we propose to extract domain-specific keywords for videos by integrating various cues from linguistic and statistical knowledge, as well as derived sound classes and characteristic visual content types. The acquisition of such salient keywords will facilitate video indexing and browsing, and significantly improve the quality of current video search engines. Experiments on four government instructional videos show that 82% of the salient keywords appear in the top 50% of the highly ranked keywords. In addition, the audiovisual cues improve precision and recall by 1.1% and 1.5% respectively.

1 Introduction

With recent advances in multimedia technology, the number of videos that are available to both general public and particular individuals or organizations is growing rapidly. This consequently creates a high demand for efficient video searching and categorization as evidenced by the emergence of various offerings for web video searching. [1]

While videos contain a rich source of audiovisual information, text-based video search is still among the most effective and widely used approaches. However, the quality of such text-based video search engines still lags behind the quality of those that search textual information like web pages. This is due to the extreme difficulty of tagging domain-specific keywords to videos. How to effectively extract domain-specific or salient keywords from video transcripts has thus become a critical and challenging issue for both the video indexing and searching communities.

Recently, with the advances in speech recognition and natural language processing technologies, systems are being developed to automatically extract keywords from video transcripts which are either transcribed from speech or obtained from closed captions. Most of these systems, however, simply treat all words equally or directly "transplant" keyword extraction techniques developed for pure text documents to the video domain without taking specific characteristics of videos into account (M. Smith and T. Kanade, 1997).

In the traditional information retrieval (IR) field, most existing methods for selecting salient keywords rely primarily on word frequency or other statistical information obtained from a collection of documents (Salton and McGill, 1983; Salton and Buckley, 1988). These techniques, however, do not work well for videos for two reasons: 1) most video transcripts are very short, as compared to a typical text collection; and 2) it is impractical to assume that there is a large video collection on a specific topic, due to the video production costs. As a result, many keywords extracted from videos using traditional IR techniques are not really content-specific, and consequently, the video search results that are returned based on these keywords are generally unsatisfactory.

In this paper, we propose a system for extracting salient or domain-specific keywords from instructional videos by exploiting joint audio, visual, and text cues. Specifically, we first apply a text-based keyword extraction system to find a set of keywords from video transcripts. Then we apply various audiovisual content analysis techniques to identify cue contexts in which domain-specific keywords are more likely to appear. Finally, we adjust the keyword salience by fusing the audio, visual and text cues together, and "discover" a set of salient keywords.

Professionally produced educational or instructional

[1] For example, see http://video.google.com and http://video.yahoo.com

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 109–112,
New York, June 2006. ©2006 Association for Computational Linguistics

videos are the main focus of this work since they are playing increasingly important roles in people's daily lives. For the system evaluation, we used training and education videos that are freely downloadable from various DHS (Department of Homeland Security) web sites. These were selected because 1) DHS has an increasing need for quickly browsing, searching and re-purposing its learning resources across its over twenty diverse agencies; 2) most DHS videos contain closed captions in compliance with federal accessibility requirements such as Section 508.

2 A Text-based Keyword Extraction System

This section describes the text-based keyword extraction system, *GlossEx*, which we developed in our earlier work (Park et al, 2002). *GlossEx* applies a hybrid method, which exploits both linguistic and statistical knowledge, to extract domain-specific keywords in a document collection. *GlossEx* has been successfully used in large-scale text analysis applications such as document authoring and indexing, back-of-book indexing, and contact center data analysis.

An overall outline of the algorithm is given below. First, the algorithm identifies candidate glossary items by using syntactic grammars as well as a set of entity recognizers. To extract more cohesive and domain-specific glossary items, it then conducts pre-nominal modifier filtering and various glossary item normalization techniques such as associating abbreviations with their full forms, and misspellings or alternative spellings with their canonical spellings. Finally, the glossary items are ranked based on their confidence values.

The confidence value of a term $T, C(T)$, is defined as

$$C(T) = \alpha * TD(T) + \beta * TC(T) \qquad (1)$$

where TD and TC denote the term domain-specificity and term cohesion, respectively. α and β are two weights which sum up to 1. The domain specificity is further defined as

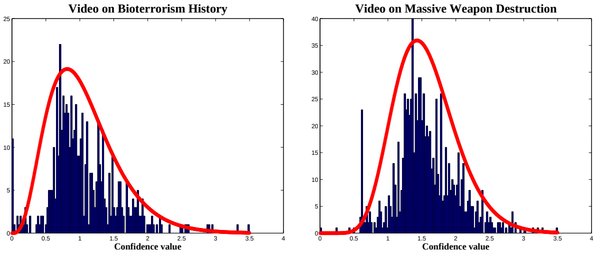
$$TD = \frac{\sum_{w_i \in T} \frac{P_d(w_i)}{P_g(w_i)}}{\mid T \mid} \qquad (2)$$

where, $\mid T \mid$ is the number of words in term T, $p_d(w_i)$ is the probability of word w_i in a domain document collection, and $p_g(w_i)$ is the probability of word w_i in a general document collection. And the term cohesion is defined as

$$TC = \frac{\mid T \mid \times f(T) \times log_{10} f(T)}{\sum_{w_i \in T} f(w_i)} \qquad (3)$$

where, $f(T)$ is the frequency of term T, and $f(w_i)$ is the frequency of a component word w_i.

Finally, *GlossEx* normalizes the term confidence values to the range of $[0, 3.5]$. Figure 1 shows the normalized distributions of keyword confidence values that we obtained from two instructional videos by analyzing their text transcripts with *GlossEx*. Superimposed on each plot is the probability density function (PDF) of a gamma distribution ($Gamma(\alpha, \gamma)$) whose two parameters are directly computed from the confidence values. As we can see, the gamma PDF fits very well with the data distribution. This observation has also been confirmed by other test videos.



(a) (b)

Figure 1: Normalized distribution of keyword saliencies for two DHS video, superimposed by Gamma PDFs.

3 Salient Keyword Extraction for Instructional Videos

In this section, we elaborate on our approach for extracting salient keywords from instructional videos based on the exploitation of audiovisual and text cues.

3.1 Characteristics of Instructional Videos

Compared to general videos, professionally produced instructional videos are usually better structured, that is, they generally contain well organized topics and subtopics due to education nature. In fact, there are certain types of production patterns that could be observed from these videos. For instance, at the very beginning section of the video, a host will usually give an overview of the main topics (as well as a list of sub-topics) that are to be discussed throughout the video. Then each individual topic or sub-topic is sequentially presented following a pre-designed order. When one topic is completed, some informational credit pages will be (optionally) displayed, followed by either some informational title pages showing the next topic, or a host introduction. A relatively long interval of music or silence that accompanies this transitional period could usually be observed in this case.

To effectively deliver the topics or materials to an audience, the video producers usually apply the following types of content presentation forms: host narration, interviews and site reports, presentation slides and information bulletins, as well as assisted content that are related with the topic under discussion. For convenience, we call the last two types as *informative text* and *linkage scene* in this work. Figure 2 shows the individual examples of video frames that contain narrator, informative text, and the linkage scene.

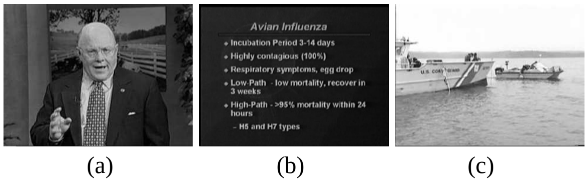

Avian Influenza
Incubation Period 3-14 days
Highly contagious (100%)
Respiratory symptoms, egg drop
Low-Path - low mortality, recover in 3 weeks
High-Path - >95% mortality within 24 hours
– H5 and H7 types
(a)
(b)
(c)

Figure 2: Three visual content types: (a) narrator, (b) informative text, and (c) linkage scene.

3.2 AudioVisual Content Analysis

This section describes our approach on mining the aforementioned content structure and patterns for instructional videos based on the analysis of both audio and visual information. Specifically, given an instructional video, we first apply an audio classification module to partition its audio track into homogeneous audio segments. Each segment is then tagged with one of the following five sound labels: speech, silence, music, environmental sound, and speech with music (Li and Dorai, 2004). The support vector machine technique is applied for this purpose.

Meanwhile, a homogeneous video segmentation process is performed which partitions the video into a series of video segments in which each segment contains content in the same physical setting. Two groups of visual features are then extracted from each segment so as to further derive its content type. Specifically, features regarding the presence of human faces are first extracted using a face detector, and these are subsequently applied to determine if the segment contains a narrator. The other feature group contains features regarding detected text blobs and sentences from the video's text overlays. This information is mainly applied to determine if the segment contains informative text. Finally, we label segments that do not contain narrators or informative text as linkage scenes. These could be an outdoor landscape, a field demonstration or indoor classroom overview. More details on this part are presented in (Li and Dorai, 2005).

The audio and visual analysis results are then integrated together to essentially assign a semantic audiovisual label to each video segment. Specifically, given a segment, we first identify its major audio type by finding the one that lasts the longest. Then, the audio and visual labels are integrated in a straightforward way to reveal its semantics. For instance, if the segment contains a narrator while its major audio type is music, it will be tagged as *narrator with music playing*. A total of fifteen possible constructs is thus generated, coming from the combination of three visual labels (narrator, informative text and linkage scene) and five sound labels (speech, silence, music, environmental sound, and speech with music).

3.3 AudioVisual and Text Cues for Salient Keyword Extraction

Having acquired video content structure and segment content types, we now extract important audiovisual cues that imply the existence of salient keywords. Specifically, we observe that topic-specific keywords are more likely appearing in the following scenarios (a.k.a *cue context*): 1) the first N_1 sentences of segments that contain narrator presentation (*i.e.* narrator with speech), or informative text with voice-over; 2) the first N_2 sentences of a new speaker (*i.e.* after a speaker change); 3) the question sentence; 4) the first N_2 sentences right after the question (i.e. the corresponding answer); and 5) the first N_2 sentences following the segments that contain silence, or informative text with music. Specifically, the first 4 cues conform with our intuition that important content subjects are more likely to be mentioned at the beginning part of narration, presentation, answers, as well as in questions; while the last cue corresponds to the transitional period between topics. Here, N_1 is a threshold which will be automatically adjusted for each segment during the process. Specifically, we set N_1 to $min(\mathrm{SS}, 3)$ where SS is the number of sentences that are overlapped with each segment. In contrast, N_2 is fixed to 2 for this work as it is only associated with sentences.

Note that currently we identify the speaker changes and question sentences by locating the signature characters (such as "$>>$" and "?") in the transcript. However, when this information is unavailable, numerous existing techniques on speaker change detection and prosody analysis could be applied to accomplish the task (Chen et al., 1998).

3.4 Keyword Salience Adjustment

Now, given each keyword (K) obtained from *GlossEx*, we recalculate its salience by considering the following three factors: 1) its original confidence value assigned by *GlossEx* ($C_{GlossEx}(K)$); 2) the frequency of the keyword occurring in the aforementioned cue context ($F_{cue}(K)$); and 3) the number of component words in the keyword ($|K|$). Specifically, we give more weight or incentive ($I(K)$) to keywords that are originally of high confidence, appear more frequently in cue contexts, and have multiple component words. Note that if keyword K does not appear in any cue contexts, its incentive value will be zero.

Figure 3 shows the detailed incentive calculation steps. Here, $mode$ and σ denote the mode and standard deviation derived from the *GlossEx* 's confidence value distribution. $MAX\_CONFIDENCE$ is the maximum confidence value used for normalization by *GlossEx*, which is set to 3.5 in this work. As we can see, the three aforementioned factors have been re-transformed into $C(K)$, $F(K)$ and $L(K)$, respectively. Please also note that we have re-adjusted the frequency of keyword K in the cue context if it is larger than 10. This intends to reduce the biased influence of a high frequency. Finally, we add a small value ϵ to $|K|$ and F_{cue} respectively in order to avoid zero values for $F(K)$ and $L(K)$. Now, we have similar value scales for $F(K)$ and $L(K)$ ($[1.09, 2.xx]$) and $C(K)$ ($[0, 2.yy]$), which is desirable.

As the last step, we boost keyword K's original salience $C_{GlossEx}(K)$ by $I(K)$.

```
if  ($C_{GlossEx}(K) >= mode$
        $C(K) = \frac{C_{GlossEx}(K)}{mode}$
else  $C(K) = \frac{C_{GlossEx}(K)}{MAX\_CONFIDENCE}$

if  ( $F_{cue}(K) > 10$)
        $F_{cue}(K) = 10 + \log_{10}(F_{cue}(K) - 10)$
$F(K) = ln(F_{cue}(K) + \epsilon)$

$L(K) = ln(|K| + \epsilon)$

$I(K) = \sigma \times C(K) \times F(K) \times L(K)$
```

Figure 3: Steps for computing incentive value for keyword K appearing in cue context

4 Experimental Results

Four DHS videos were used in the experiment, which contain diverse topics ranging from bio-terrorism history, weapons of mass destruction, to school preparation for terrorism. The video length also varies a lot from 30 minutes to 2 hours. Each video also contains a variety of sub-topics. Video transcripts were acquired by extracting the closed captions with our own application.

To evaluate system performance, we compare the keywords generated from our system against the human-generated gold standard. Note that for this experiment, we only consider nouns and noun phrases as keywords. To collect the ground truth, we invited a few human evaluators, showed them the four test videos, and presented them with all candidate keywords extracted by *GlossEx*. We then asked them to label all keywords that they considered to be domain-specific, which is guidelined by the following question: "*would you be satisfied if you get this video when you use this keyword as a search term*?".

Table 1 shows the number of candidate keywords and manually labeled salient keywords for all four test videos. As we can see, approximately 50% of candidate keywords were judged to be domain-specific by humans. Based on this observation, we selected the top 50% of highly ranked keywords based on the adjusted salience, and examined their presence in the pool of salient keywords for each video. As a result, an average of 82% of salient keywords were identified within these top 50% of re-ranked keywords. In addition, the audiovisual cues improve precision and recall by 1.1% and 1.5% respectively.

videos	v_1	v_2	v_3	v_4
no. of candidate keywords	477	934	1303	870
no. of salient keywords	253	370	665	363
ratio of salient keywords	53%	40%	51%	42%

Table 1: The number of candidate and manually labeled salient keywords in the four test videos

5 Conclusion and Future Work

We described a mutimodal feature-based system for extracting salient keywords from instructional videos. The system utilizes a richer set of information cues which not only include linguistic and statistical knowledge but also sound classes and characteristic visual content types that are available to videos. Experiments conducted on the DHS videos have shown that incorporating multimodal features for extracting salient keywords from videos is useful.

Currently, we are performing more sophisticated experiments on different ways to exploit additional audio-visual cues. There is also room for improving the calculation of the incentive values of keywords. Our next plan is to conduct an extensive comparison between *GlossEx* and the proposed scheme.

References

Y. Park, R. Byrd and B. Boguraev. 2002. *Automatic Glossary Extraction: Beyond Terminology Identification*. Proc. of the 19th International Conf. on Computational Linguistics (COLING02), pp 772–778.

Y. Li and C. Dorai. 2004 *SVM-based Audio Classification for Instructional Video Analysis*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04).

Y. Li and C. Dorai. 2005 *Video frame identification for learning media content understanding*. IEEE International Conference on Multimedia & Expo (ICME'05).

M. Smith and T. Kanade. 1997 *Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques*. IEEE Computer Vision and Pattern Recognition, pp. 775-781.

G. Salton and J. McGill 1983. *Introduction to modern information Retrieval*. . New York: McGraw-Hill.

G. Salton and C. Buckley 1988. *Term-Weighting Approaches in Automatic Text Retrieval*. Information Processing & Management, 24 (5), 513-523.

S. Chen and P. Gopalakrishnan 1998. *Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion*. Proc. of DARPA Broadcast News Transcription and Understanding Workshop.

Exploiting Variant Corpora for Machine Translation

Michael Paul[†‡] and **Eiichiro Sumita**[†‡]
† National Institute of Information and Communications Technology
‡ ATR Spoken Language Communication Research Labs
Hikaridai 2-2-2, Keihanna Science City, 619-0288 Kyoto
{Michael.Paul,Eiichiro.Sumita}@{nict.go.jp,atr.jp}

Abstract

This paper proposes the usage of *variant corpora*, i.e., parallel text corpora that are equal in meaning but use different ways to express content, in order to improve corpus-based machine translation. The usage of multiple training corpora of the same content with different sources results in variant models that focus on specific linguistic phenomena covered by the respective corpus. The proposed method applies each variant model separately resulting in multiple translation hypotheses which are selectively combined according to statistical models. The proposed method outperforms the conventional approach of merging all variants by reducing translation ambiguities and exploiting the strengths of each variant model.

1 Introduction

Corpus-based approaches to machine translation (MT) have achieved much progress over the last decades. Despite a high performance on average, these approaches can often produce translations with severe errors. Input sentences featuring linguistic phenomena that are not sufficiently covered by the utilized models cannot be translated accurately.

This paper proposes to use multiple *variant corpora*, i.e., parallel text corpora that are equal in meaning, but use different vocabulary and grammatical constructions in order to express the same content. Using training corpora of the same content with different sources result in translation models that focus on specific linguistic phenomena, thus reducing translation ambiguities compared to models trained on a larger corpus obtained by merging all variant corpora. The proposed method applies each variant model separately to an input sentence resulting in multiple translation hypotheses. The best translation is selected according to statistical models. We show that the combination of variant translation models is effective and outperforms not only all single variant models, but also is superior to translation models trained on the union of all variant corpora.

In addition, we extend the proposed method to multi-engine MT. Combining multiple MT engines can boost the system performance further by exploiting the strengths of each MT engine. For each variant, all MT engines are trained on the same corpus and used in parallel to translate the input. We first select the best translation hypotheses created by all MT engines trained on the same variant and then verify the translation quality of the translation hypotheses selected for each variant.

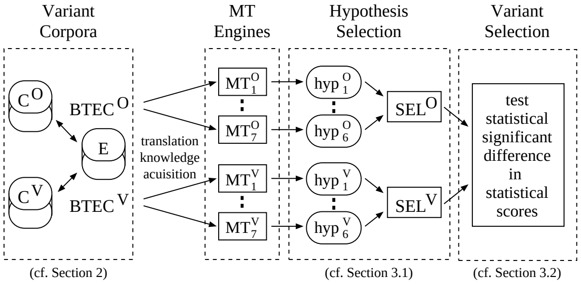


Figure 1: System outline

The outline of the proposed system is given in Figure 1. For the experiments described in this paper we are using two variants of a parallel text corpus for Chinese (C) and English (E) from the travel domain (cf. Section 2). These variant corpora are used to acquire the translation knowledge for seven corpus-based MT engines. The method to select the best translation hypotheses of MT engines trained on the same variant is described in Section 3.1. Finally, the selected translations of different variants are combined according to a statistical significance test as described in Section 3.2. The effectiveness of the proposed method is verified in Section 4 for

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 113–116,
New York, June 2006. ©2006 Association for Computational Linguistics

the Chinese-English translation task of last year's IWSLT[1] evaluation campaign.

2 Variant Corpora

The *Basic Travel Expressions Corpus* (BTEC) is a collection of sentences that bilingual travel experts consider useful for people going to or coming from another country and cover utterances in travel situations (Kikui et al., 2003). The original Japanese-English corpus consists of 500K of aligned sentence pairs whereby the Japanese sentences were also translated into Chinese.

In addition, parts of the original English corpus were translated separately into Chinese resulting in a variant corpus comprising 162K CE sentence pairs. Details of both, the original (*BTEC^O*) and the variant (*BTEC^V*) corpus, are given in Table 1, where *word token* refers to the number of words in the corpus and *word type* refers to the vocabulary size.

Table 1: Statistics of variant corpora

corpus	lang	sentence count		avg	word	word
		total	unique	len	tokens	types
BTEC^O	C	501,809	299,347	6.8	3,436,750	40,645
	E	501,809	344,134	8.3	4,524,703	21,832
BTEC^V	C	162,320	97,512	7.1	1,302,761	14,222
	E	162,320	96,988	7.5	1,367,981	9,795

Only 4.8% of the sentences occured in both corpora and only 68.1% of the BTEC^V vocabulary was covered in the BTEC^O corpus.

The comparison of both corpora revealed further that each variant closely reflects the linguistic structure of the source language which was used to produce the Chinese translations of the respective data sets. The differences between the BTEC^O and BTEC^V variants can be categorized into:

(1) **literalness:** BTEC^O sentences are translated on the basis of their meaning and context resulting in freer translations compared to the BTEC^V sentences which are translated more literally;

(2) **syntax:** The degree of literalness also has an impact on the syntactic structure like word order variations (C^V sentences reflect closely the word order of the corresponding English sentences) or the sentence type (*question* vs. *imperative*);

(3) **lexical choice:** Alternations in lexical choice also contribute largely to variations between the corpora. Moreover, most of the pronouns found in the English sentences are translated explicitly in the C^V sentences, but are omitted in C^O;

(4) **orthography:** Orthographic differences especially for proper nouns (*Kanji* vs. *transliteration*) and numbers (*numerals* vs. *spelling-out*).

3 Corpus-based Machine Translation

The differences in variant corpora directly effect the translation quality of corpus-based MT approaches. Simply merging variant corpora for training increases the coverage of linguistic phenomena by the obtained translation model. However, due to an increase in translation ambiguities, more erroneous translations might be generated.

In contrast, the proposed method trains separately MT engines on each variant focusing on linguistic phenomena covered in the respective corpus. If specific linguistic phenomena are not covered by a variant corpus, the translation quality of the respective output is expected to be significantly lower.

Therefore, we first judge the translation quality of all translation hypotheses created by MT engines trained on the same variant corpus by testing statistical significant differences in the statistical scores (cf. Section 3.1). Next, we compare the outcomes of the statistical significance test between the translation hypotheses selected for each variant in order to identify the variant that fits best the given input sentence (cf. Section 3.2).

3.1 Hypothesis Selection

In order to select the best translation among outputs generated by multiple MT systems, we employ an SMT-based method that scores MT outputs by using multiple language (LM) and translation model (TM) pairs trained on different subsets of the training data. It uses a statistical test to check whether the obtained TM·LM scores of one MT output are significantly higher than those of another MT output (Akiba et al., 2002). Given an input sentence, m translation hypotheses are produced by the element MT engines, whereby n different TM·LM scores are assigned to each hypothesis. In order to check whether the highest scored hypothesis is significantly better then the other MT outputs, a multiple comparison test based on the Kruskal-Wallis test is used. If one of the MT outputs is significantly better, this output is selected.

[1] http://penance.is.cs.cmu.edu/iwslt2005

Otherwise, the output of the MT engine that performs best on a develop set is selected.

3.2 Variant Selection

In order to judge which variant should be selected for the translation of a given input sentence, the outcomes of the statistical significance test carried out during the hypothesis selection are employed.

The hypothesis selection method is applied for each variant separately, i.e., the BTEC^O corpus is used to train multiple statistical model pairs (SEL^O) and the best translation (MT$^O_{SEL}$) of the set of translation hypotheses created by the MT engines trained on the BTEC^O corpus is selected. Accordingly, the SEL^V models are trained on the BTEC^V corpus and applied to select the best translation (MT$^V_{SEL}$) of the MT outputs trained on the BTEC^V corpus. In addition, the SEL^O models were used in order to verify whether a significant difference can be found for the translation hypothesis MT$^V_{SEL}$, and, vice versa, the SEL^V models were applied to MT$^O_{SEL}$.

The outcomes of the statistical significance tests are then compared. If a significant difference between the statistical scores based on one variant, but not for the other variant is obtained, the significantly better hypothesis is selected as the output. However, if a significant difference could be found for both or none of the variants, the translation hypothesis produced by the MT engine that performs best on a develop set is selected.

4 Experiments

The effectiveness of the proposed method is verified for the CE translation task (500 sentences) of last year's IWSLT evaluation campaign. For the experiments, we used the four *statistical* (SMT) and three *example-based* (EBMT) MT engines described in detail in (Paul et al., 2005).

For evaluation, we used the BLEU metrics, which calculates the geometric mean of n-gram precision for the MT outputs found in reference translations (Papineni et al., 2002). Higher BLEU scores indicate better translations.

4.1 Performance of Element MT Engines

Table 2 summarizes the results of all element MT engines trained on the BTEC^O and BTEC^V corpora. The result show that the SMT engines outperform the EBMT engines whereby the best performing system is marked with bold-face.

Table 2: BLEU evaluation of element MT engines

SMT	BTEC^O	BTEC^V	EBMT	BTEC^O	BTEC^V
MT$_1$	0.4020	0.4633	MT$_5$	0.2908	0.3445
MT$_2$	0.4474	0.4595	MT$_6$	0.2988	0.4100
MT$_3$	**0.5342**	**0.5110**	MT$_7$	0.0002	0.0074
MT$_4$	0.3575	0.4460			

However, depending on the variant corpus used to train the MT engines, quite different system performances are achieved. Most of the element MT engines perform better when trained on the smaller BTEC^V corpus indicating that the given test set is not covered well by the BTEC^O corpus.

4.2 Effects of Hypothesis Selection

The performance of the hypothesis selection method (SEL) is summarized in Table 3 whereby the obtained gain relative to the best element MT engine is given in parentheses. In addition, we performed an "oracle" translation experiment in order to investigate in an upper boundary for the method. Each input sentence was translated by all element MT engines and the translation hypothesis with the lowest word error rate[2] relative to the reference translations was output as the translation, i.e., the ORACLE system simulates an optimal selection method according to an objective evaluation criterion.

Table 3: BLEU evaluation of hypothesis selection

MT engine	BTEC^O	BTEC^V
SEL	**0.5409** (+ 0.7%)	**0.5470** (+ 3.6%)
ORACLE	0.6385 (+10.4%)	0.6502 (+13.9%)

MT engine	BTEC$^{O \cup V}$
SEL	**0.4648** (–7.0%)
ORACLE	0.6969 (+16.3%)

The results show that the selection method is effective for both variant corpora whereby a larger gain is achieved for BTEC^V. However, the ORACLE results indicate that the method fails to tap the full potential of the element MT engines.

In addition, we trained the statistical models of the hypothesis selection method on the corpus obtained

[2] The *word error rate* (WER) is an objective evaluation measures that, in contrast to BLEU, can be applied on sentence-level. It penalizes edit operations for the translation output against reference translations.

by merging all variant corpora (BTEC$^{O \cup V}$). Despite the larger amount of training data, the BLEU score decreases drastically which shows that an increase in training data not necessarily leads to improved translation quality. Moreover, the ORACLE selection applied to all translation hypotheses based on the BTEC^O as well as the BTEC^V corpus indicates that both variants can contribute significantly in order to improve the overall system performance.

4.3 Effects of Variant Selection

The effects of combining selected variant hypotheses by testing whether significant differences in statistical scores were obtained are summarized in Table 4. The variant selection method is applied to the translation outputs of each element MT engine (MT$_j^O$ $\parallel$ MT$_j^V$) as well as the selected translation hypotheses (MT$_{SEL}^O$ $\parallel$ MT$_{SEL}^V$). The gain of the proposed variant selection method relative the best element MT output based on a single variant corpus is given in parentheses.

Table 4: BLEU evaluation of variant selection

MT engine		BTEC^O $\parallel$ BTEC^V	
SMT	MT$_1^O$ $\parallel$ MT$_1^V$	0.5010	(+ 3.8%)
	MT$_2^O$ $\parallel$ MT$_2^V$	0.4847	(+ 2.5%)
	MT$_3^O$ $\parallel$ MT$_3^V$	0.5594	(+ 2.5%)
	MT$_4^O$ $\parallel$ MT$_4^V$	0.4733	(+ 2.7%)
EBMT	MT$_5^O$ $\parallel$ MT$_5^V$	0.3863	(+ 4.2%)
	MT$_6^O$ $\parallel$ MT$_6^V$	0.4338	(+ 2.4%)
	MT$_7^O$ $\parallel$ MT$_7^V$	0.0181	(+10.7%)
	MT$_{SEL}^O$ $\parallel$ **MT**$_{SEL}^V$	**0.5765**	(+ 4.2%)

The results show that the variant selection method is effective for all element MT engines. The highest BLEU score is achieved for **MT**$_{SEL}^O$ $\parallel$ **MT**$_{SEL}^V$ gaining 4.2% in BLEU score. Moreover, the proposed method outperforms the hypothesis selection method based on the merged corpus BTEC$^{O \cup V}$ by 11.2% in BLEU score.

A comparison of the proposed method with the best performing system (C-STAR data track, BLEU=0.5279) of the IWSLT 2005 workshop showed that our system outperforms the top-ranked system gaining 4.8% in BLEU score.

5 Conclusion

This paper proposed the usage of variant corpora to improve the translation quality of a multi-engine-based approach to machine translation. The element MT engines were used to translate the same input whereby the best translation was selected according to statistical models. A test on the significance of differences between statistical scores judging the translation quality of a given hypothesis was exploited to identify the model that fits the input sentence best and the respective translation hypothesis was selected as the translation output.

The proposed method was evaluated on the CE translation task of the IWSLT 2005 workshop. The results showed that the proposed method achieving a BLEU score of 0.5765 outperformed not only all element MT engines (gaining 3.6% in BLEU score), but also a selection method using a larger corpus obtained from merging all variant corpora (gaining 11.2% in BLEU score) due to less ambiguity in the utilized models. In addition, the proposed method also outperformed the best MT system (C-STAR data track) of the IWSLT 2005 workshop gaining 4.8% in BLEU score.

Further investigations should analyze the characteristics of the variant corpora in more detail and focus on the automatic identification of specific linguistic phenomena that could be helpful to measure how good an input sentence is covered by a specific model. This would allow us to select the most adequate variant beforehand, thus reducing computational costs and improving the system performance. This would also enable us to cluster very large corpora according to specific linguistic phenomena, thus breaking down the full training corpus to consistent subsets that are easier to manage and that could produce better results.

References

K. Papineni et al. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the 40th ACL*, pages 311–318.

Y. Akiba et al. 2002. Using language and translation models to select the best among outputs from multiple MT systems. In *Proc. of COLING*, pages 8–14.

G. Kikui et al. 2003. Creating corpora for speech-to-speech translation. In *Proc. of EUROSPEECH03*, pages 381–384.

M. Paul et al. 2005. Nobody is Perfect: ATR's Hybrid Approach to Spoken Language Translation. In *Proc. of the IWSLT*, pages 55–62.

Quantitative Methods for Classifying Writing Systems

Gerald Penn
University of Toronto
10 King's College Rd.
Toronto M5S 3G4, Canada
gpenn@cs.toronto.edu

Travis Choma
Cognitive Science Center Amsterdam
Sarphatistraat 104
1018 GV Amsterdam, Netherlands
travischoma@gmail.com

Abstract

We describe work in progress on using quantitative methods to classify writing systems according to Sproat's (2000) classification grid using unannotated data. We specifically propose two quantitative tests for determining the type of phonography in a writing system, and its degree of logography, respectively.

1 Background

If you understood all of the world's languages, you would still not be able to read many of the texts that you find on the world wide web, because they are written in non-Roman scripts that have been arbitrarily encoded for electronic transmission in the absence of an accepted standard. This very modern nuisance reflects a dilemma as ancient as writing itself: the association between a language as it is spoken and the language as it is written has a sort of internal logic to it that we can comprehend, but the conventions are different in every individual case — even among languages that use the same script, or between scripts used by the same language. This conventional association between language and script, called a *writing system*, is indeed reminiscent of the Saussurean conception of language itself, a conventional association of meaning and sound, upon which modern linguistic theory is based.

Despite linguists' necessary reliance upon writing to present and preserve linguistic data, however, writing systems were a largely neglected corner of linguistics until the 1960s, when Gelb (1963) presented the first classification of writing systems. Now known as the *Gelb teleology*, this classification viewed the variation we see among writing systems, particularly in the size of linguistic "chunks" represented by an individual character or unit of writing (for simplicity, referred to here as a *grapheme*), along a linear, evolutionary progression, beginning with the pictographic forerunners of writing, proceeding through "primitive" writing systems such as Chinese and Egyptian hieroglyphics, and culminating in alphabetic Greek and Latin.

While the linear and evolutionary aspects of Gelb's teleology have been rejected by more recent work on the classification of writing systems, the admission that more than one dimension may be necessary to characterize the world's writing systems has not come easily. The ongoing polemic between Sampson (1985) and DeFrancis (1989), for example, while addressing some very important issues in the study of writing systems,[1] has been confined exclusively to a debate over which of several arboreal classifications of writing is more adequate.

Sproat (2000)'s classification was the first multidimensional one. While acknowledging that other dimensions may exist, Sproat (2000) arranges writing systems along the two principal dimensions of *Type of Phonography* and *Amount of Logography*, both of which will be elaborated upon below. This is the departure point for our present study.

Our goal is to identify quantitative methods that

[1] These include what, if anything, separates true writing systems from other more limited written forms of communication, and the psychological reality of our classifications in the minds of native readers.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 117–120,
New York, June 2006. ©2006 Association for Computational Linguistics

			Type of Phonography			
	Consonantal	Polyconsonantal	Alphabetic		Core Syllabic	Syllabic
←Amount of Logography	W. Semitic		English, Greek, Korean, Devanagari	Pahawh Hmong	Linear B	Modern Yi
	Perso-Aramaic					
						Chinese
		Egyptian			Sumerian, Mayan, Japanese	

Figure 1: Sproat's writing system classification grid (Sproat, 2000, p. 142).

can assist in the classification of writing systems. On the one hand, these methods would serve to verify or refute proposals such as Sproat's (2000, p. 142) placement of several specific writing systems within his grid (Figure 1) and to properly place additional writing systems, but they could also be used, at least corroboratively, to argue for the existence of more appropriate or additional dimensions in such grids, through the demonstration of a pattern being consistently observed or violated by observed writing systems. The holy grail in this area would be a tool that could classify entirely unknown writing systems to assist in attempts at archaeological decipherment, but more realistic applications do exist, particularly in the realm of managing on-line document collections in heterogeneous scripts or writing systems.

No previous work exactly addresses this topic. None of the numerous descriptive accounts that catalogue the world's writing systems, culminating in Daniels and Bright's (1996) outstanding reference on the subject, count as quantitative. The one computational approach that at least claims to consider archaeological decipherment (Knight and Yamada, 1999), curiously enough, assumes an alphabetic and purely phonographic mapping of graphemes at the outset, and applies an EM-style algorithm to what is probably better described as an interesting variation on learning the "letter-to-sound" mappings that one normally finds in text analysis for text-to-speech synthesizers. The cryptographic work in the great wars of the early 20th century applied statistical reasoning to military communications, although this too is very different in character from deciphering a naturally developed writing system.

2 Type of Phonography

Type of phonography, as it is expressed in Sproat's grid, is not a continuous dimension but a discrete choice by graphemes among several different phonographic encodings. These characterize not only the size of the phonological "chunks" encoded by a single grapheme (progressing left-to-right in Figure 1 roughly from small to large), but also whether vowels are explicitly encoded (poly/consonantal vs. the rest), and, in the case of vocalic syllabaries, whether codas as well as onsets are encoded (core syllabic vs. syllabic). While we cannot yet discriminate between all of these phonographic aspects (arguably, they are different dimensions in that a writing system may select a value from each one independently), size itself can be reliably estimated from the number of graphemes in the underlying script, or from this number in combination with the tails of grapheme distributions in representative documents. Figure 2, for example, graphs the frequencies of the grapheme types witnessed among the first 500 grapheme tokens of one document sampled from an on-line newspaper website in each of 8 different writing systems plus an Egyptian hieroglyphic document from an on-line repository. From left to right, we see the alphabetic and consonantal (small chunks) scripts, followed by the polyconsonantal Egyptian hieroglyphics, followed by core syllabic Japanese, and then syllabic Chinese. Korean was classified near Japanese because its Unicode representation atomically encodes the multisegment syllabic complexes that characterize most Hangul writing. A segmental encoding would appear closer to English.

3 Amount of Logography

Amount of logography is rather more difficult. Roughly, logography is the capacity of a writing system to associate the symbols of a script directly with the meanings of specific words rather than indirectly through their pronunciations. No one to our knowledge has proposed any justification for whether logography should be viewed continuously or discretely. Sproat (2000) believes that it is continuous, but acknowledges that this belief is more impressionistic than factual. In addition, it appears, according to Sproat's (2000) discussion that amount or degree of logography, whatever it is, says something about the relative frequency with which graphemic tokens are used semantically, rather than about the properties of individual graphemes in isolation. English, for example, has a very low degree of logography, but it does have logographic graphemes and graphemes that can be used in a logographic aspect. These include numerals (with or without phonographic complements as in "3^{rd}," which distinguishes "3" as "three" from "3" as "third"), dollar signs, and arguably some common abbreviations as "etc." By contrast, type of phonography predicts a property that holds of every individual grapheme — with few exceptions (such as symbols for word-initial vowels in CV syllabaries), graphemes in the same writing system are marching to the same drum in their phonographic dimension.

Another reason that amount of logography is difficult to measure is that it is not entirely independent of the type of phonography. As the size of the phonological units encoded by graphemes increases, at some point a threshold is crossed wherein the unit is about the size of a word or another meaning-bearing unit, such as a bound morpheme. When this happens, the distinction between phonographic and logographic uses of such graphemes becomes a far more intensional one than in alphabetic writing systems such as English, where the boundary is quite clear. Egyptian hieroglyphics are well known for their use of *rebus signs*, for example, in which highly pictographic graphemes are used not for the concepts denoted by the pictures, but for concepts with words pronounced like the word for the depicted concept. There are very few writing systems indeed where the size of the phonological unit is word-sized and yet the writing system is still mostly phonographic;[2] it could be argued that the distinction simply does not exist (see Section 4).

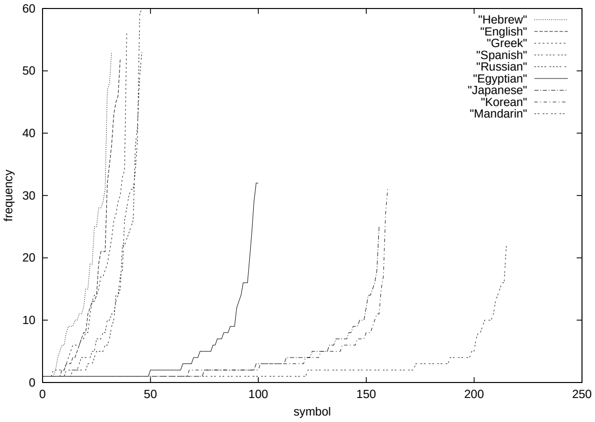


Figure 2: Grapheme distributions in 9 writing systems. The symbols are ordered by inverse frequency to separate the heads of the distributions better. The left-to-right order of the heads is as shown in the key.

Nevertheless, one can distinguish *pervasive* semantical use from *pervasive* phonographic use. We do not have access to electronically encoded Modern Yi text, so to demonstrate the principle, we will use English text re-encoded so that each "grapheme" in the new encoding represents three consecutive graphemes (breaking at word boundaries) in the underlying natural text. We call this *trigraph English*, and it has no (intensional) logography. The principle is that, if graphemes are pervasively used in their semantical respect, then they will "clump" semantically just like words do. To measure this clumping, we use *sample correlation coefficients*. Given two random variables, X and Y, their correlation is given by their covariance, normalized by their sample standard deviations:

$$\begin{array}{l} corr(X,Y) = \frac{cov(X,Y)}{s(X) \cdot s(Y)} \\ cov(X,Y) = \frac{1}{n-1}\Sigma_{0 \leq i,j \leq n}(x_i - \mu_i)(y_j - \mu_j) \\ s(X) = \sqrt{\frac{1}{n-1}\Sigma_{0 \leq i \leq n}(x_i - \mu)^2} \end{array}$$

For our purposes, each grapheme type is treated as a variable, and each document represents an observation. Each cell of the matrix of correlation coefficients then tells us the strength of the correlation between two grapheme types. For trigraph English, part of the correlation matrix is shown in Figure 3. Part of the correlation matrix for Mandarin

[2] Modern Yi (Figure 1) is one such example, although the history of Modern Yi is more akin to that of a planned language than a naturally evolved semiotic system.

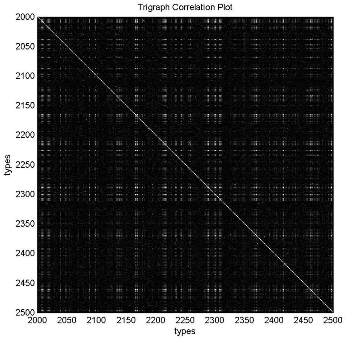


Figure 3: Part of the trigraph-English correlation matrix.

Chinese, which has a very high degree of logography, is shown in Figure 4. For both of the plots in our example, counts for 2500 grapheme types were obtained from 1.63 million tokens of text (for English, trigraphed Brown corpus text, for Chinese, GB5-encoded text from an on-line newspaper).

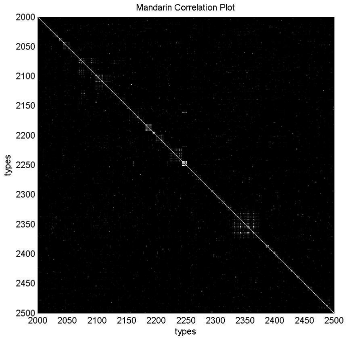


Figure 4: Part of the Mandarin Chinese correlation matrix.

By adding the absolute values of the correlations over these matrices (normalized for number of graphemes), we obtain a measure of the extent of the correlation. Pervasive semantic clumping, which would be indicative of a high degree of logography, corresponds to a small extent of correlation — in other words the correlation is pinpointed at semantically related logograms, rather than smeared over semantically orthogonal phonograms. In our example, these sums were repeated for several 2500-type samples from among the approximately 35,000 types in the trigraph English data, and the approximately 4,500 types in the Mandarin data. The average sum for trigraph English was 302,750 whereas for Mandarin Chinese it was 98,700. Visually, this difference is apparent in that the trigraph English matrix is “brighter” than the Mandarin one. From this we should conclude that Mandarin Chinese has a higher degree of logography than trigraph English.

4 Conclusion

We have proposed methods for independently measuring the type of phonography and degree of logography from unannotated data as a means of classifying writing systems. There is more to understanding how a writing system works than these two dimensions. Crucially, the direction in which texts should be read, the so-called *macroscopic organization* of typical documents, is just as important as determining the functional characteristics of individual graphemes.

Our experiments with quantitative methods for classification, furthermore, have led us to a new understanding of the differences between Sproat’s classification grid and earlier linear attempts. While we do not accept Gelb’s teleological interpretation, we conjecture that there is a linear variation in how individual writing systems behave, even if they can be classified according to multiple dimensions. Modern Yi stands as a single, but questionable, counterexample to this observation, and for it to be visible in Sproat’s grid (with writing systems arranged along only the diagonal), one would need an objective and verifiable means of discriminating between consonantal and vocalic scripts. This remains a topic for future consideration.

References

P. Daniels and W. Bright. 1996. *The World’s Writing Systems*. Oxford.

J. DeFrancis. 1989. *Visible Speech: The Diverse Oneness of Writing Systems*. University of Hawaii.

I. Gelb. 1963. *A Study of Writing*. Chicago, 2nd ed.

K. Knight and K. Yamada. 1999. A computational approach to deciphering unknown scripts. In *Proc. of ACL Workshop on Unsupervised Learning in NLP*.

G. Sampson. 1985. *Writing Systems*. Stanford.

R. Sproat. 2000. *A Computational Theory of Writing Systems*. Cambridge University Press.

Computational Modelling of Structural Priming in Dialogue

David Reitter, Frank Keller, Johanna D. Moore
dreitter | keller | jmoore @ inf.ed.ac.uk
School of Informatics
University of Edinburgh
United Kingdom

Abstract

Syntactic priming effects, modelled as increase in repetition probability shortly after a use of a syntactic rule, have the potential to improve language processing components. We model priming of syntactic rules in annotated corpora of spoken dialogue, extending previous work that was confined to selected constructions. We find that speakers are more receptive to priming from their interlocutor in task-oriented dialogue than in sponaneous conversation. Low-frequency rules are more likely to show priming.

1 Introduction

Current dialogue systems overlook an interesting fact of language-based communication. Speakers tend to repeat their linguistic decisions rather than making them from scratch, creating *entrainment* over time. Repetition is evident not just on the obvious lexical level: *syntactic* choices depend on preceding ones in a way that can be modelled and, ultimately, be leveraged in parsing and language generation. The statistical analysis in this paper aims to make headway towards such a model.

Recently, priming phenomena[1] have been exploited to aid automated processing, for instance in automatic speech recognition using cache models, but only recently have attempts been made at using them in parsing (Charniak and Johnson, 2005). In natural language generation, repetition can be used to increase the alignment of human and computers. A surface-level approach is possible by biasing the n-gram language model used to select the output string from a variety of possible utterances (Brockmann et al., 2005).

Priming effects are common and well known. For instance, speakers access lexical items more quickly after a semantically or phonologically similar prime. Recent work demonstrates large effects for particular synonymous alternations (e.g., active vs. passive voice) using traditional laboratory experiments with human subjects (Bock, 1986; Branigan et al., 2000). In this study, we look at the effect from a computational perspective, that is, we assume some form of parsing and syntax-driven generation components. While previous studies singled out syntactic phenomena, we assume a phrase-structure grammar where all syntactic rules may receive priming. We use large-scale corpora, which reflect the realities of natural interaction, where limited control exists over syntax and the semantics of the utterances. Thus, we quantify priming for the general case in the realistic setting provided by corpus based experiments. As a first hypothesis, we predict that after a a syntactic rule occurs, it is more likely to be repeated shortly than a long time afterwards.

From a theoretical perspective, priming opens a peephole into the architecture of the human language faculty. By identifying units in which priming occurs, we can pinpoint the structures used in processing. Also, priming may help explain the ease with which humans engange in conversations.

This study is interested in the differences relevant to systems implementing language-based human-

[1] The term *priming* refers to a process that influences linguistic decision-making. An instance of priming occurs when a syntactic structure or lexical item giving evidence of a linguistic choice (*prime*) influences the recipient to make the same decision, i.e. re-use the structure, at a later choice-point (*target*).

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 121–124,
New York, June 2006.

computer interaction. Often, HCI is a means for user and system to jointly plan or carry out a task. Thus, we look at repetition effects in task-oriented dialogue. A recent psychological perspective models *Interactive Alignment* between speakers (Pickering and Garrod, 2004), where mutual understanding about task and situation depends on lower-level priming effects. Under the model, we expect priming effects to be stronger when a task requires high-level alignment of situation models.

2 Method

2.1 Dialogue types

We examined two corpora. *Switchboard* contains 80,000 utterances of *spontaneous spoken conversations* over the telephone among randomly paired, North American speakers, syntactically annotated with phrase-structure grammar (Marcus et al., 1994). *The HCRC Map Task* corpus comprises more than 110 dialogues with a total of $20,400$ utterances (Anderson et al., 1991). Like Switchboard, HCRC Map Task is a corpus of spoken, two-person dialogue in English. However, Map Task contains *task-oriented dialogue*: interlocutors work together to achieve a task as quickly and efficiently as possible. Subjects were asked to give each other directions with the help of a map. The interlocutors are in the same room, but have separate, slightly different maps and are unable to see each other's maps.

2.2 Syntactic repetitions

Both corpora are annotated with phrase structure trees. Each tree was converted into the set of phrase structure productions that license it. This allows us to identify the repeated use of rules. Structural priming would predict that a rule *(target)* occurs more often shortly after a potential *prime* of the same rule than long afterwards – any repetition at great distance is seen as coincidental. Therefore, we can correlate the probability of repetition with the elapsed time (DIST) between prime and target.

We considered very pair of two equal syntactic rules up to a predefined maximal distance to be a potential case of priming-enhanced production. If we consider priming at distances $1 \dots n$, each rule instance produces up to n data points. Our binary response variable indicates whether there is a prime for the target between $n - 0.5$ and $n + 0.5$ seconds before the target. As a prime, we see the invocation of the same rule. Syntactic repetitions resulting from lexical repetition and repetitions of unary rules are excluded. We looked for repetitions within windows (DIST) of $n = 15$ seconds (Section 3.1).

Without priming, one would expect that there is a constant probability of syntactic repetition, no matter the distance between prime and target. The analysis tries to reject this null hypothesis and show a correlation of the effect size with the type of corpus used. We expect to see the syntactic priming effect found experimentally should translate to more cases for shorter repetition distances, since priming effects usually decay rapidly (Branigan et al., 1999).

The target utterance is included as a random factor in our model, grouping all 15 measurements of all rules of an utterance as *repeated measurements*, since they depend on the same target rule occurrence or at least on other other rules in the utterance, and are, thus, partially inter-dependent.

We distinguish *production-production priming* within (PP) and *comprehension-production priming* between speakers (CP), encoded in the factor ROLE. Models were estimated on joint data sets derived from both corpora, with a factor SOURCE included to discriminate the two dialogue types.

Additionally, we build a model estimating the effect of the raw frequency of a particular syntactic rule on the priming effect (FREQ). This is of particular interest for priming in applications, where a statistical model will, all other things equal, prefer the more frequent linguistic choice; recall for competing low-frequency rules will be low.

2.3 Generalized Linear Mixed Effect Regression

In this study, we built generalized linear mixed effects regression models (GLMM). In all cases, a rule instance *target* is counted as a repetition at distance d iff there is an utterance *prime* which contains the same rule, and *prime* and *target* are d units apart. GLMMs with a logit-link function are a form of *logistic regression*.[2]

[2] We trained our models using Penalized Quasi-Likelihood (Venables and Ripley, 2002). We will not generally give classical R^2 figures, as this metric is not appropriate to such GLMMs. The below experiments were conducted on a sample of 250,000

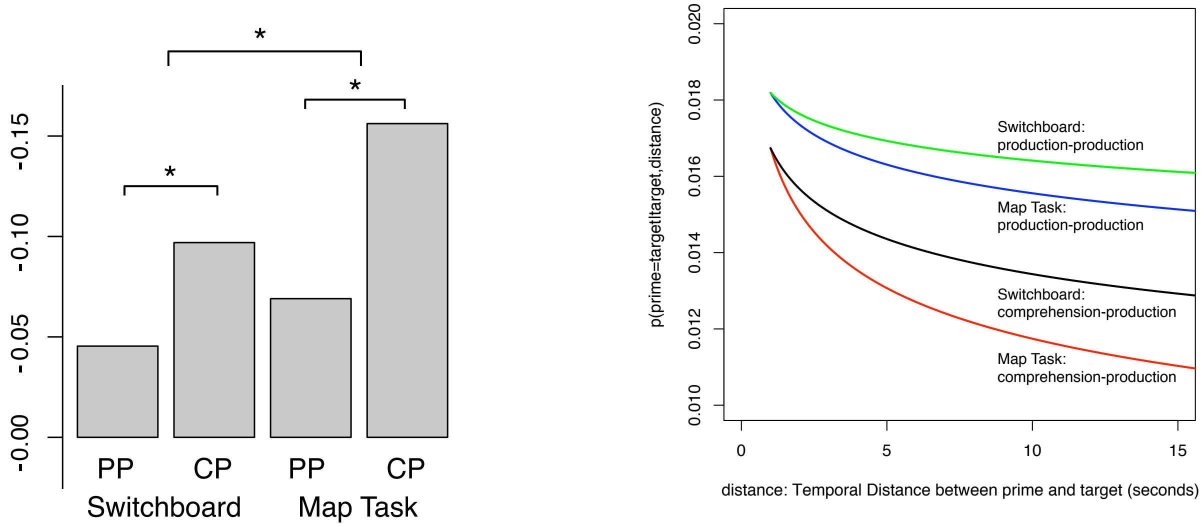


Figure 1: Left: Estimated priming strength (repetition probability decay rate) for Switchboard and Map Task, for within-speaker (PP) and between-speaker (CP) priming. Right: Fitted model for the development of repetition probability (y axis) over time (x axis, in seconds). Here, decay (slope) is the relevant factor for priming strength, as shown on the left. These are derived from models without FREQ.

Regression allows us not only to show that priming exists, but it allows us to predict the decline of repetition probability with increasing distance between prime and target and depending on other variables. If we see priming as a form of pre-activation of syntactic nodes, it indicates the decay rate of pre-activation. Our method quantifies priming and correlates the effect with secondary factors.

3 Results

3.1 Task-oriented and spontaneous dialogue

Structural repetition between speakers occured in both corpora and its probability decreases logarithmically with the distance between prime and target.

Figure 1 provides the model for the influence of the four factorial combinations of ROLE and SOURCE on priming (left) and the development of repetition probability at increasing distance (right). SOURCE=Map Task has an interaction effect on the priming decay $ln(\text{DIST})$, both for PP priming ($\beta = -0.024, t = -2.0, p < 0.05$) and for CP priming ($\beta = -0.059, t = -4.0, p < 0.0005$). (Lower coefficients indicate more decay, hence more priming.)

In both corpora, we find positive priming effects. However, PP priming is stronger, and CP priming is much stronger in Map Task.

The choice of corpus exhibits a marked interaction with priming effect. Spontaneous conversation shows significantly less priming than task-oriented dialogue. We believe this is not a side-effect of varying grammar size or a different syntactic entropy in the two types of dialogue, since we examine the *decay of repetition probability* with increasing distance (interactions with DIST), and not the overall probability of chance repetition (intercepts / main effects except DIST).

3.2 Frequency effects

An additional model was built which included $ln(\text{FREQ})$ as a predictor that may interact with the effect coefficient for $ln(\text{DIST})$.

$ln(\text{FREQ})$ is inversely correlated with the priming effect (Paraphrase: $\beta_{lnDist} = -1.05, \beta_{lnDist:lnFreq} = 0.54$, Map Task: $\beta_{lnDist} = -2.18, \beta_{lnDist:lnFreq} = 0.35$, all $p < 0.001$). Priming weakens with higher (logarithmic) frequency of a syntactic rule.

data points per corpus.

4 Discussion

Evidence from Wizard-of-Oz experiments (with systems simulated by human operators) have shown that users of dialogue systems strongly align their syntax with that of a (simulated) computer (Branigan et al., 2003). Such an effect can be leveraged in an application, provided there is a priming model interfacing syntactic processing.

We found evidence of priming in general, that is, when we assume priming of each phrase structure rule. The priming effects decay quickly and non-linearly, which means that a dialogue system would best only take a relatively short preceding context into account, e.g., the previous few utterances.

An important consideration in the context of dialogue systems is whether user and system collaborate on solving a task, such as booking tickets or retrieving information. Here, syntactic priming *between* human speakers is strong, so a system should implement it. In other situations, systems do not have to use a unified syntactic architecture for parsing and generation, but bias their output on previous system utterances, and possibly improve parsing by looking at previously recognized inputs.

The fact that priming is more pronounced *within* (PP) a speaker suggests that optimizing parsing and generation separately is the most promising avenue in either type of dialogue system.

One explanation for this lies in a reduced cognitive load of spontaneous, everyday conversation. Consequently, the more accessible, highly-frequent rules prime less.

In task-oriented dialogue, speakers need to produce a common situation model. Interactive Alignment Model argues that this process is aided by syntactic priming. In support of this model, we find more priming in task-oriented dialogue.[3]

5 Conclusions

Syntactic priming effects are reliably present in dialogue even in computational models where the full range of syntactic rules is considered instead of selected constructions with known strong priming.

This is good news for dialogue systems, which tend to be task-oriented. Linguistically motivated systems can possibly exploit the user's tendency to repeat syntactic structures by anticipating repetition. Future systems may also align their output with their recognition capabilities and actively align with the user to signal understanding. Parsers and realizers in natural language generation modules may make the most of priming if they respect important factors that influence priming effects, such as task-orientation of the dialogue and frequency of the syntactic rule.

Acknowledgements

The authors would like to thank Amit Dubey, Roger Levy and Martin Pickering. The first author's work is supported by a grant from the Edinburgh Stanford Link.

References

A. Anderson, M. Bader, E. Bard, E. Boyle, G. M. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. Thompson, and R. Weinert. 1991. The HCRC Map Task corpus. *Language and Speech*, 34(4):351–366.

J. Kathryn Bock. 1986. Syntactic persistence in language production. *Cognitive Psychology*, 18:355–387.

Holly P. Branigan, Martin J. Pickering, and Alexandra A. Cleland. 1999. Syntactic priming in language production: Evidence for rapid decay. *Psychonomic Bulletin and Review*, 6(4):635–640.

Holly P. Branigan, Martin J. Pickering, and Alexandra A. Cleland. 2000. Syntactic co-ordination in dialogue. *Cognition*, 75:B13–25.

Holly P. Branigan, Martin J. Pickering, Jamie Pearson, Janet F. McLean, and Clifford Nass. 2003. Syntactic alignment between computers and people: the role of belief about mental states. In *Proceedings of the Twenty-fifth Annual Conference of the Cognitive Science Society*.

Carsten Brockmann, Amy Isard, Jon Oberlander, and Michael White. 2005. Modelling alignment for affective dialogue. In *Workshop on Adapting the Interaction Style to Affective Factors at the 10th International Conference on User Modeling (UM-05)*. Edinburgh, UK.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. 43th ACL*.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *Proc. ARPA Human Language Technology Workshop*.

Martin J. Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–225.

David Reitter, Johanna D. Moore, and Frank Keller. 2006. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*.

William N. Venables and Brian D. Ripley. 2002. *Modern Applied Statistics with S. Fourth Edition.* Springer.

[3] For a more detailed analysis from the perspective of interactive alignment, see Reitter et al. (2006).

Story Segmentation of Brodcast News in English, Mandarin and Arabic

Andrew Rosenberg
Computer Science Department
Columbia University
New York City, N.Y. 10027
amaxwell@cs.columbia.edu

Julia Hirschberg
Computer Science Department
Columbia University
New York City, N.Y. 10027
julia@cs.columbia.edu

Abstract

In this paper, we present results from a Broadcast News story segmentation system developed for the SRI NIGHTINGALE system operating on English, Arabic and Mandarin news shows to provide input to subsequent question-answering processes. Using a rule-induction algorithm with automatically extracted acoustic and lexical features, we report success rates that are competitive with state-of-the-art systems on each input language. We further demonstrate that features useful for English and Mandarin are **not** discriminative for Arabic.

1 Introduction

Broadcast News (BN) shows typically include multiple unrelated stories, interspersed with anchor presentations of headlines and commercials. Transitions between each story are frequently marked by changes in speaking style, speaker participation, and lexical choice. Despite receiving a considerable amount of attention through the Spoken Document Retrieval (SDR), Topic Detection and Tracking (TDT), and Text Retrieval Conference: Video (TRECVID) research programs, automatic detection of story boundaries remains an elusive problem. State-of-the-art story segmentation error rates on English and Mandarin BN remain fairly high and Arabic is largely unstudied. The NIGHTINGALE system searches a diverse news corpus to return answers to user queries. For audio sources, the identification of story boundaries is crucial, to segment material to be searched and to provide interpretable results to the user.

2 Related work

Previous approaches to story segmentation have largely focused lexical features, such as word similarily (Kozima, 1993), cue phrases (Passonneau and Litman, 1997), cosine similarity of lexical windows (Hearst, 1997; Galley et al., 2003), and adaptive language modeling (Beeferman et al., 1999). Segmentation of stories in BN have included some acoustic features (Shriberg et al., 2000; Tür et al., 2001). Work on non-English BN, generally use this combination of lexical and acoustic measures, such as (Wayne, 2000; Levow, 2004) on Mandarin. And (Palmer et al., 2004) report results from feature selection experiments that include Arabic sources, though they do not report on accuracy. TRECVID has also identified visual cues to story segmentation of video BN (cf. (Hsu et al., 2004; Hsieh et al., 2003; Chaisorn et al., 2003; Maybury, 1998)).

3 The NIGHTINGALE Corpus

The training data used for NIGHTINGALE includes the TDT-4 and TDT5 corpora (Strassel and Glenn, 2003; Strassel et al., 2004). TDT-4 includes newswire text and broadcast news audio in English, Arabic and Mandarin; TDT-5 contains only text data, and is therefore not used by our system. The TDT-4 audio corpus includes 312.5 hours of English Broadcast News from 450 shows, 88.5 hours of Arabic news from 109 shows, and 134 hours of Mandarin broadcasts from 205 shows. This material was drawn from six English news shows – ABC “World News Tonight”, CNN “Headline News”, NBC “Nightly News”, Public Radio International “The World”, MS-NBC “News with Brian Williams”, and Voice of America, English three Mandarin newscasts — China National Radio, China Television System, and Voice of America, Mandarin Chinese — and two Arabic newscasts — Nile TV and Voice of America, Modern Standard Arabic. All of these shows aired between Oct. 1, 2000 and Jan. 31, 2001.

4 Our Features and Approach

Our story segmentation system procedure is essentially one of binary classification, trained on a variety of acoustic and lexical cues to the presence or absence of story boundaries in BN. Our classifier was trained using the JRip machine learning al-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 125–128,
New York, June 2006. ©2006 Association for Computational Linguistics

gorithm, a Java implementation of the RIPPER algorithm of (Cohen, 1995).[1] All of the cues we use are automatically extracted. We use as input to our classifier three types of automatic annotation produced by other components of the NIGHTINGALE system, speech recognition (ASR) transcription, speaker diarization, sentence segmentation. Currently, we assume that story boundaries occur only at these hypothesized sentence boundaries. For our English corpus, this assumption is true for only 47% of story boundaries; the average reference story boundary is 9.88 words from an automatically recognized sentence boundary[2]. This errorful input immediately limits our overall performance.

For each such hypothesized sentence boundary, we extract a set of features based on the previous and following hypothesized sentences. The classifier then outputs a prediction of whether or not this sentence boundary coincides with a story boundary. The features we use for story boundary prediction are divided into three types: lexical, acoustic and speaker-dependent.

The value of even errorful lexical information in identifying story boundaries has been confirmed for many previous story segmentation systems (Beeferman et al., 1999; Stokes, 2003)). We include some previously-tested types of lexical features in our own system, as well as identifying our own 'cue-word' features from our training corpus. Our lexical features are extracted from ASR transcripts produced by the NIGHTINGALE system. They include lexical similarity scores calculated from the TextTiling algorithm.(Hearst, 1997), which determines the lexical similarity of blocks of text by analyzing the cosine similarity of a sequence of sentences; this algorithm tests the likelihood of a topic boundary between blocks, preferring locations between blocks which have minimal lexical similarity. For English, we stem the input before calculating these features, using an implementation of the Porter stemmer (Porter, 1980); we have not yet attempted to identify root forms for Mandarin or Arabic. We also calculate scores from (Galley et al., 2003)'s LCseg method, a TextTiling-like approach which weights the cosine-similarity of a text window by an additional measure of its component LEXICAL CHAINS, repetitions of stemmed content words. We also identify 'cue-words' from our training data that we find to be significantly more likely (determined by χ^2) to occur at story boundaries within a window preceeding or following a story boundary. We include as features the number of such words observed within 3, 5, 7 and 10 word windows before and after the candidate sentence boundary. For English, we include the number of pronouns contained in the sentence, on the assumption that speakers would use more pronouns at the end of stories than at the beginning. We have not yet obtained reliable part-of-speech tagging for Arabic or Mandarin. Finally, for all three languages, we include features that represent the sentence length in words, and the relative sentence position in the broadcast.

Acoustic/prosodic information has been shown to be indicative of topic boundaries in both spontaneous dialogs and more structured speech, such as, broadcast news (cf. (Hirschberg and Nakatani, 1998; Shriberg et al., 2000; Levow, 2004)). The acoustic features we extract include, for the current sentence, the minimum, maximum, mean, and standard deviation of F0 and intensity, and the median and mean absolute slope of F0 calculated over the entire sentence. Additionally, we compute the first-order difference from the previous sentence of each of these. As a approximation of each sentence's speaking rate, we include the ratio of voiced 10ms frames to the total number of frames in the sentence. These acoustic values were extracted from the audio input using Praat speech analysis software(Boersma, 2001). Also, using the phone alignment information derived from the ASR process, we calculate speaking rate in terms of the number of vowels per second as an additional feature. Under the hypothesis that topic-ending sentences may exhibit some additional phrase-final lenghthening, we compare the length of the sentence-final vowel and of the sentence-final rhyme to average durations for that vowel and rhyme for the speaker, where speaker identify is available from the NIGHTINGALE diarization component; otherwise we use unnormalized values.

We also use speaker identification information from the diarization component to extract some fea-

[1] JRip is implemented in the Weka (Witten et al., 1999) machine learning environment.

[2] For Mandarin and Arabic respectively, true for 69% and 62% with the average distance between sentence and story boundary of 1.97 and 2.91 words.

tures indicative of a speaker's participation in the broadcast as a whole. We hypothesize that participants in a broadcast may have different roles, such as an anchor providing transitions between stories and reporters beginning new stories (Barzilay et al., 2000) and thus that speaker identity may serve as a story boundary indicator. To capture such information, we include binary features answering the questions: "Is the speaker preceeding this boundary the first speaker in the show?", "Is this the first time the speaker has spoken in this broadcast?", "The last time?", and "Does a speaker boundary occur at this sentence boundary?". Also, we include the percentage of sentences in the broadcast spoken by the current speaker.

We assumed in the development of this system that the source of the broadcast is known, specifically the source language and the show identity (e. g. ABC "World News Tonight", CNN "Headline News"). Given this information, we constructed different classifiers for each show. This type of source-specific modeling was shown to improve performance by Tür (2001).

5 Results and Discussion

We report the results of our system on English, Mandarin and Arabic in Table 5. All results use show-specific modeling, which consistently improved our results across all metrics, reducing errors by between 10% and 30%. In these tables, we report the F-measure of identifying the precise location of a story boundary as well as three metrics designed specifically for this type of segmentation task: the pk metric (Beeferman et al., 1999), *WindowDiff* (Pevzner and Hearst, 2002) and C_{seg} (P_{seg} = 0.3) (Doddington, 1998). All three are derived from the pk metric (Beeferman et al., 1999), and for all, lower values imply better performance. For each of these three metrics we let $k = 5$, as prescribed in (Beeferman et al., 1999).

In every system, the best peforming results are achieved by including all features from the lexical, acoustic and speaker-dependent feature sets. Across all languages, our precision–and false alarm rates–are better than recall–and miss rates. We believe that inserting erroneous story boundaries will lead to more serious downstream errors in anaphora resolution and summarization than a boundary omission will. Therefore, high precision is more important than high recall for a helpful story segmentation system. In the English and Mandarin systems, the lexical and acoustic feature sets perform similarly, and combine to yield improved results. However, on the Arabic data, the acoustic feature set performs quite poorly, suggesting that the use of vocal cues to topic transitions may be fundamentally different in Arabic. Moreover, these differences are not simply differences of degree or direction. Rather, the acoustic indicators of topic shifts in English and Mandarin are, simply, not discriminative when applied to Arabic. This difference may be due to the style of Arabic newscasts or to the language itself. Across configurations, we find that the inclusion of features derived from automatic speaker identification (feature set S), errorful as it is, significantly improves performance. This improvement is particularly pronounced on the Mandarin material; in China News Radio broadcasts, story boundaries are very strongly correlated with speaker transitions.

It is difficult to determine how well our system performs against state-of-the-art story segmentation. There are no comparable results for the TDT-4 corpus. On the English TDT-2 corpus, (Shriberg et al., 2000) report a C_{seg} score of 0.1438. While our score of .0670 is half that, we hesitate to conclude that our system is significantly better than this system; since the (Shriberg et al., 2000) results are based on a word-level segmentation, the discrepancy may be influenced by the disparate datasets as well as the performance of the two systems. On CNN and Reuters stories from the TDT-1 corpus, (Stokes, 2003) report a Pk score of 0.25 and a WD score of 0.253. Our Pk score is better than this on TDT-4, while our WD score is worse. (Chaisorn et al., 2003) report an F-measure of 0.532 using only audio-based features on the TRECVID 2003 corpus , which is higher than our system, however, this allows for "correct" boundaries to fall within 5 seconds of reference boundaries. (Franz et al., 2000) present a system which achieves C_{seg} scores of 0.067 and Mandarin BN and 0.081 on English audio in TDT-3. This suggests that their system may be better than ours on Mandarin, and worse on English, although we trained and tested on different corpora. Finally, we are unaware of any reported story segmentation results on Arabic BN.

Table 1: TDT-4 segmentation results. (L=lexical feature set, A=acoustic, S=speaker-dependent)

	English				Mandarin				Arabic			
	F1(p,r)	Pk	WD	C_{seg}	F1(p,r)	Pk	WD	C_{seg}	F1(p,r)	Pk	WD	C_{seg}
L+A+S	.421(.67,.31)	.194	.318	.0670	.592(.73,.50)	.179	.245	.0679	.300(.65,.19)	.264	.353	.0850
A+S	.346(.65,.24)	.220	.349	.0721	.586(.72,.49)	.178	.252	.0680	.0487(.81,.03)	.333	.426	.0999
L+S	.342(.66,.23)	.231	.362	.074	.575(.72,.48)	.200	.278	.0742	.285(.68,.18)	.286	.372	.0884
L+A	.319(.66,.21)	.240	.376	.0787	.294(.72,.18)	.277	.354	.0886	.284(.64,.18)	.257	.344	.0851
L	.257(.68,.16)	.261	.399	.0840	.226(.74,.13)	.309	.391	.0979	.286(.68,.18)	.283	.349	.0849
A	.194(.63,.11)	.271	.412	.0850	.252(.72,.18)	.291	.377	.0904	.0526(.81,.03)	.332	.422	.0996

6 Conclusion

In this paper we have presented results of our story boundary detection procedures on English, Mandarin, and Arabic Broadcast News from the TDT-4 corpus. All features are obtained automatically, except for the identity of the news show and the source language, information which is, however, available from the data itself, and could be automatically obtained. Our performance on TDT-4 BN appears to be better than previous work on earlier corpora of BN for English, and slightly worse than previous efforts on Mandarin, again for a different corpus. We believe our Arabic results to be the first reported evaluation for BN in that language. One important observation from our study is that acoustic/prosodic features that correlate with story boundaries in English and in Mandarin, do not correlate with Arabic boundaries. Our further research will adress the study of vocal cues to segmentation in Arabic BN.

Acknowledgments

This research was partially supported by the Defese Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023.

References

R. Barzilay, M. Collins, J. Hirschberg, and S. Whittaker. 2000. The rules behind roles: Identifying speaker role in radio broadcasts. In *AAAI/IAAI*, 679–684.

D. Beeferman, A. Berger, and J. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 31:177–210.

P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9-10):341–345.

L. Chaisorn, T. Chua, C. Koh, Y. Zhao, H. Xu, H. Feng, and Q. Tian. 2003. A two-level multi-modeal approach for story segmentation of large news video corpus. In *TRECVID*.

W. Cohen. 1995. Fast effective rule induction. In *Machine Learning: Proc. of the Twelfth International Conference*, 115–123.

G. Doddington. 1998. The topic detection and tracking phase 2 (tdt2) evaluation plan. In *Proccedings DARPA Broadcast News Transcription and Understanding Workshop*, 223–229.

M. Franz, J. S. McCarley, T. Ward, and W. J. Zhu. 2000. Segmentation and detection at ibm: Hybrid statstical models and two-tiered clustering. In *Proc. of TDT-3 Workshop*.

M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. 2003. Discourse segmentation of multi-party conversation. In *41st Annual Meeting of ACL*, 562–569.

M. A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

J. Hirschberg and C. Nakatani. 1998. Acoustic indicators of topic segmentation. In *Proc. of ICSLP*, 1255–1258.

J. H. Hsieh, C. H. Wu, and K. A. Fung. 2003. Two-stage story segmentation and detection on broadcast news using genetic algorithm. In *Proc. of the 2003 ISCA Workshop on Multilingual Spoken Document Retrieval (MSDR2003)*, 55–60.

W. Hsu, L. Kennedy, C. W. Huang, S. F. Chang, C. Y. Lin, and G. Iyengar. 2004. News video story segmentation using fusion of multi-level multi-modal features in trecvid 2003. In *ICASSP*.

H. Kozima. 1993. Text segmentation based on similarity between words. In *31st Annual Meeting of the ACL*, 286–288.

G. A. Levow. 2004. Assessing prosodic and text features for segmentation of mandarin broadcast news. In *HLT-NAACL*.

M. T. Maybury. 1998. Discourse cues for broadcast news segmentation. In *COLING-ACL*, 819–822.

D. D. Palmer, M. Reichman, and E. Yaich. 2004. Feature selection for trainable multilingual broadcast news segmentation. In *HLT/NAACL*.

R. J. Passonneau and D. J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Liunguistics*, 23(1):103–109.

L. Pevzner and M. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür. 2000. Prosody based automatic segmentation of speech into sentences and topics. *Speech Comm.*, 32(1-2):127–154.

N. Stokes. 2003. Spoken and written news story segmentation using lexical chains. In *Proc. of the Student Workshop at HLT-NAACL2003*, 49–53.

S. Strassel and M. Glenn. 2003. Creating the annotated tdt-4 y2003 evaluation corpus. http://www.nist.gov/speech/tests/tdt/tdt2003/papers/ldc.ppt.

S. Strassel, M. Glenn, and J. Kong. 2004. Creating the tdt5 corpus and 2004 evalutation topics at ldc. http://www.nist.gov/speech/tests/tdt/tdt2004/papers/LDC-TDT5.ppt.

G. Tür, D. Hakkani-Tür, A. Stolcke, and E. Shriberg. 2001. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27:31–57.

C. L. Wayne. 2000. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *LREC*, 1487–1494.

I. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. Cunningham. 1999. Weka: Practical machine learning tools and techniques with java implementation. In *ICONIP/ANZIIS/ANNES*, 192–196.

Parser Combination by Reparsing

Kenji Sagae and **Alon Lavie**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{sagae,alavie@cs.cmu.edu}

Abstract

We present a novel parser combination scheme that works by reparsing input sentences once they have already been parsed by several different parsers. We apply this idea to dependency and constituent parsing, generating results that surpass state-of-the-art accuracy levels for individual parsers.

1 Introduction

Over the past decade, remarkable progress has been made in data-driven parsing. Much of this work has been fueled by the availability of large corpora annotated with syntactic structures, especially the Penn Treebank (Marcus et al., 1993). In fact, years of extensive research on training and testing parsers on the Wall Street Journal (WSJ) corpus of the Penn Treebank have resulted in the availability of several high-accuracy parsers.

We present a framework for combining the output of several different accurate parsers to produce results that are superior to those of each of the individual parsers. This is done in a two stage process of *reparsing*. In the first stage, *m* different parsers analyze an input sentence, each producing a syntactic structure. In the second stage, a parsing algorithm is applied to the original sentence, taking into account the analyses produced by each parser in the first stage. Our approach produces results with accuracy above those of the best individual parsers on both dependency and constituent parsing of the standard WSJ test set.

2 Dependency Reparsing

In dependency reparsing we focus on unlabeled dependencies, as described by Eisner (1996). In this scheme, the syntactic structure for a sentence with *n* words is a dependency tree representing head-dependent relations between pairs of words.

When *m* parsers each output a set of dependencies (forming *m* dependency structures) for a given sentence containing *n* words, the dependencies can be combined in a simple word-by-word voting scheme, where each parser votes for the head of each of the *n* words in the sentence, and the head with most votes is assigned to each word. This very simple scheme guarantees that the final set of dependencies will have as many votes as possible, but it does not guarantee that the final voted set of dependencies will be a well-formed dependency tree. In fact, the resulting graph may not even be connected. Zeman & Žabokrtský (2005) apply this dependency voting scheme to Czech with very strong results. However, when the constraint that structures must be well-formed is enforced, the accuracy of their results drops sharply.

Instead, if we reparse the sentence based on the output of the *m* parsers, we can maximize the number of votes for a well-formed dependency structure. Once we have obtained the *m* initial dependency structures to be combined, the first step is to build a graph where each word in the sentence is a node. We then create weighted directed edges between the nodes corresponding to words for which dependencies are obtained from each of the initial structures.[1] In cases where more than one dependency structure indicates that an edge should be created, the corresponding weights are simply added. As long as at least one of the *m* initial structures is a well-formed dependency structure, the directed graph created this way will be connected.

[1] Determining the weights is discussed in section 4.1.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 129–132,
New York, June 2006. ©2006 Association for Computational Linguistics

Once this graph is created, we reparse the sentence using a dependency parsing algorithm such as, for example, one of the algorithms described by McDonald et al. (2005). Finding the optimal dependency structure given the set of weighted dependencies is simply a matter of finding the maximum spanning tree (MST) for the directed weighted graph, which can be done using the Chu-Liu/Edmonds directed MST algorithm (Chu & Liu, 1965; Edmonds, 1967). The maximum spanning tree maximizes the votes for dependencies given the constraint that the resulting structure must be a tree. If projectivity (no crossing branches) is desired, Eisner's (1996) dynamic programming algorithm (similar to CYK) for dependency parsing can be used instead.

3 Constituent Reparsing

In constituent reparsing we deal with labeled constituent trees, or phrase structure trees, such as those in the Penn Treebank (after removing traces, empty nodes and function tags). The general idea is the same as with dependencies. First, *m* parsers each produce one parse tree for an input sentence. We then use these *m* initial parse trees to guide the application of a parse algorithm to the input.

Instead of building a graph out of words (nodes) and dependencies (edges), in constituent reparsing we use the *m* initial trees to build a weighted parse chart. We start by decomposing each tree into its constituents, with each constituent being a 4-tuple [*label, begin, end, weight*], where *label* is the phrase structure type, such as NP or VP, *begin* is the index of the word where the constituent starts, *end* is the index of the word where the constituent ends plus one, and *weight* is the weight of the constituent. As with dependencies, in the simplest case the weight of each constituent is simply 1.0, but different weighting schemes can be used. Once the initial trees have been broken down into constituents, we put all the constituents from all of the *m* trees into a single list. We then look for each pair of constituents *A* and *B* where the *label*, *begin*, and *end* are identical, and merge *A* and *B* into a single constituent with the same *label*, *begin*, and *end*, and with *weight* equal to the *weight* of *A* plus the *weight* of *B*. Once no more constituent mergers are possible, the resulting constituents are placed on a standard parse chart, but where the constituents in the chart do not contain back-pointers indicating what smaller constituents they contain. Building the final tree amounts to determining these back-pointers. This can be done by running a bottom-up chart parsing algorithm (Allen, 1995) for a weighted grammar, but instead of using a grammar to determine what constituents can be built and what their weights are, we simply constrain the building of constituents to what is already in the chart (adding the weights of constituents when they are combined). This way, we perform an exhaustive search for the tree that represents the heaviest combination of constituents that spans the entire sentence as a well-formed tree.

A problem with simply considering all constituents and picking the heaviest tree is that this favors recall over precision. Balancing precision and recall is accomplished by discarding every constituent with weight below a threshold *t* before the search for the final parse tree starts. In the simple case where each constituent starts out with weight 1.0 (before any merging), this means that a constituent is only considered for inclusion in the final parse tree if it appears in at least *t* of the *m* initial parse trees. Intuitively, this should increase precision, since we expect that a constituent that appears in the output of more parsers to be more likely to be correct. By changing the threshold *t* we can control the precision/recall tradeoff.

Henderson and Brill (1999) proposed two parser combination schemes, one that picks an entire tree from one of the parsers, and one that, like ours, builds a new tree from constituents from the initial trees. The latter scheme performed better, producing remarkable results despite its simplicity. The combination is done with a simple majority vote of whether or not constituents should appear in the combined tree. In other words, if a constituent appears at least *(m + 1)/2* times in the output of the *m* parsers, the constituent is added to the final tree. This simple vote resulted in trees with f-score significantly higher than the one of the best parser in the combination. However, the scheme heavily favors precision over recall. Their results on WSJ section 23 were 92.1 precision and 89.2 recall (90.61 f-score), well above the most accurate parser in their experiments (88.6 f-score).

4 Experiments

In our dependency parsing experiments we used unlabeled dependencies extracted from the Penn

Treebank using the same head-table as Yamada and Matsumoto (2003), using sections 02-21 as training data and section 23 as test data, following (McDonald et al., 2005; Nivre & Scholz, 2004; Yamada & Matsumoto, 2003). Dependencies extracted from section 00 were used as held-out data, and section 22 was used as additional development data. For constituent parsing, we used the section splits of the Penn Treebank as described above, as has become standard in statistical parsing research.

4.1 Dependency Reparsing Experiments

Six dependency parsers were used in our combination experiments, as described below.

The deterministic shift-reduce parsing algorithm of (Nivre & Scholz, 2004) was used to create two parsers[2], one that processes the input sentence from left-to-right (LR), and one that goes from right-to-left (RL). Because this deterministic algorithm makes a single pass over the input string with no back-tracking, making decisions based on the parser's state and history, the order in which input tokens are considered affects the result. Therefore, we achieve additional parser diversity with the same algorithm, simply by varying the direction of parsing. We refer to the two parsers as LR and RL.

The deterministic parser of Yamada and Matsumoto (2003) uses an algorithm similar to Nivre and Scholz's, but it makes several successive left-to-right passes over the input instead of keeping a stack. To increase parser diversity, we used a version of Yamada and Matsumoto's algorithm where the direction of each of the consecutive passes over the input string alternates from left-to-right and right-to-left. We refer to this parser as LRRL.

The large-margin parser described in (McDonald et al., 2005) was used with no alterations. Unlike the deterministic parsers above, this parser uses a dynamic programming algorithm (Eisner, 1996) to determine the best tree, so there is no difference between presenting the input from left-to-right or right-to-left.

Three different weight configurations were considered: (1) giving all dependencies the same weight; (2) giving dependencies different weights, depending only on which parser generated the dependency; and (3) giving dependencies different weights, depending on which parser generated the dependency, and the part-of-speech of the dependent word. Option 2 takes into consideration that parsers may have different levels of accuracy, and dependencies proposed by more accurate parsers should be counted more heavily. Option 3 goes a step further, attempting to capitalize on the specific strengths of the different parsers.

The weights in option 2 are determined by computing the accuracy of each parser on the held-out set (WSJ section 00). The weights are simply the corresponding parser's accuracy (number of correct dependencies divided by the total number of dependencies). The weights in option 3 are determined in a similar manner, but different accuracy figures are computed for each part-of-speech.

Table 1 shows the dependency accuracy and root accuracy (number of times the root of the dependency tree was identified correctly divided by the number of sentences) for each of the parsers, and for each of the different weight settings in the reparsing experiments (numbered according to their descriptions above).

System	Accuracy	Root Acc.
LR	91.0	92.6
RL	90.1	86.3
LRRL	89.6	89.1
McDonald	90.9	94.2
Reparse dep 1	91.8	96.0
Reparse dep 2	92.1	95.9
Reparse dep 3	**92.7**	**96.6**

Table 1: Dependency accuracy and root accuracy of individual dependency parsers and their combination under three different weighted reparsing settings.

4.2 Constituent Reparsing Experiments

The parsers that were used in the constituent reparsing experiments are: (1) Charniak and Johnson's (2005) reranking parser; (2) Henderson's (2004) synchronous neural network parser; (3) Bikel's (2002) implementation of the Collins (1999) model 2 parser; and (4) two versions of Sagae and Lavie's (2005) shift-reduce parser, one using a maximum entropy classifier, and one using support vector machines.

Henderson and Brill's voting scheme mentioned in section 3 can be emulated by our reparsing approach by setting all weights to 1.0 and t to $(m + 1)/2$, but better results can be obtained by setting appropriate weights and adjusting the precision/recall tradeoff. Weights for different types of

[2] Nivre and Scholz use memory based learning in their experiments. Our implementation of their parser uses support vector machines, with improved results.

constituents from each parser can be set in a similar way to configuration 3 in the dependency experiments. However, instead of measuring accuracy for each part-of-speech tag of dependents, we measure precision for each non-terminal label.

The parameter *t* is set using held-out data (from WSJ section 22) and a simple hill-climbing procedure. First we set *t* to *(m + 1)/2* (which heavily favors precision). We then repeatedly evaluate the combination of parsers, each time decreasing the value of *t* (by 0.01, say). We record the values of *t* for which precision and recall were closest, and for which f-score was highest.

Table 2 shows the accuracy of each individual parser and for three reparsing settings. Setting 1 is the emulation of Henderson and Brill's voting. In setting 2, *t* is set for balancing precision and recall. In setting 3, *t* is set for highest f-score.

System	Precision	Recall	F-score
Charniak/Johnson	91.3	90.6	91.0
Henderson	90.2	89.1	89.6
Bikel (Collins)	88.3	88.1	88.2
Sagae/Lavie (a)	86.9	86.6	86.7
Sagae/Lavie (b)	88.0	87.8	87.9
Reparse 1	**95.1**	88.5	91.6
Reparse 2	91.8	**91.9**	91.8
Reparse 3	93.2	91.0	**92.1**

Table 2: Precision, recall and f-score of each constituent parser and their combination under three different reparsing settings.

5 Discussion

We have presented a reparsing scheme that produces results with accuracy higher than the best individual parsers available by combining their results. We have shown that in the case of dependencies, the reparsing approach successfully addresses the issue of constructing high-accuracy well-formed structures from the output of several parsers. In constituent reparsing, held-out data can be used for setting a parameter that allows for balancing precision and recall, or increasing f-score. By combining several parsers with f-scores ranging from 91.0% to 86.7%, we obtain reparsed results with a 92.1% f-score.

References

Allen, J. (1995). *Natural Language Understanding* (2nd ed.). Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc.

Bikel, D. (2002). Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of HLT2002*. San Diego, CA.

Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd meeting of the Association for Computational Linguistics*. Ann Arbor, MI.

Chu, Y. J., & Liu, T. H. (1965). On the shortest arborescence of a directed graph. *Science Sinica*(14), 1396-1400.

Edmonds, J. (1967). Optimum branchings. *Journal of Research of the National Bureau of Standards*(71B), 233-240.

Eisner, J. (1996). Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the International Conference on Computational Linguistics (COLING'96)*. Copenhagen, Denmark.

Henderson, J. (2004). Discriminative training of a neural network statistical parser. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*. Barcelona, Spain.

Henderson, J., & Brill, E. (1999). Exploiting diversity in natural language processing: combining parsers. In *Proceedings of the Fourth Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Marcus, M. P., Santorini, B., & Marcinkiewics, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics, 19.*

McDonald, R., Pereira, F., Ribarov, K., & Hajic, J. (2005). Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of the Conference on Human Language Technologies/Empirical Methods in Natural Language Processing (HLT-EMNLP)*. Vancouver, Canada.

Nivre, J., & Scholz, M. (2004). Deterministic dependency parsing of English text. In *Proceedings of the 20th International Conference on Computational Linguistics* (pp. 64-70). Geneva, Switzerland.

Sagae, K., & Lavie, A. (2005). A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technologies.* Vancouver, Canada.

Yamada, H., & Matsumoto, Y. (2003). Statistical dependency analysis using support vector machines. In *Proceedings of the Eighth International Workshop on Parsing Technologies*. Nancy, France.

Zeman, D., & Žabokrtský, Z. (2005). Improving Parsing Accuracy by Combining Diverse Dependency Parsers. In *Proceedings of the International Workshop on Parsing Technologies*. Vancouver, Canada.

Using Phrasal Patterns to Identify Discourse Relations

Manami Saito
Nagaoka University of Technology
Niigata, JP 9402188
saito@nlp.nagaokaut.ac.jp

Kazuhide Yamamoto
Nagaoka University of Technology
Niigata, JP 9402188
yamamoto@fw.ipsj.or.jp

Satoshi Sekine
New York University
New York, NY 10003
sekine@cs.nyu.edu

Abstract

This paper describes a system which identifies discourse relations between two successive sentences in Japanese. On top of the lexical information previously proposed, we used phrasal pattern information. Adding phrasal information improves the system's accuracy 12%, from 53% to 65%.

1 Introduction

Identifying discourse relations is important for many applications, such as text/conversation understanding, single/multi-document summarization and question answering. (Marcu and Echihabi 2002) proposed a method to identify discourse relations between text segments using Naïve Bayes classifiers trained on a huge corpus. They showed that lexical pair information extracted from massive amounts of data can have a major impact.

We developed a system which identifies the discourse relation between two successive sentences in Japanese. On top of the lexical information previously proposed, we added phrasal pattern information. A phrasal pattern includes at least three phrases (bunsetsu segments) from two sentences, where function words are mandatory and content words are optional. For example, if the first sentence is "X should have done Y" and the second sentence is "A did B", then we found it very likely that the discourse relation is CONTRAST (89% in our Japanese corpus).

2 Discourse Relation Definitions

There have been many definitions of discourse relation, for example (Wolf 2005) and (Ichikawa 1987) in Japanese. We basically used Ichikawa's classes and categorized 167 cue phrases in the ChaSen dictionary (IPADIC, Ver.2.7.0), as shown in Table 1. Ambiguous cue phrases were categorized into multiple classes. There are 7 classes, but the OTHER class will be ignored in the following experiment, as its frequency is very small.

Table 1. Discourse relations

Discourse relation	Examples of cue phrase (English translation)	Freq. in corpus [%]
ELABORATION	and, also, then, moreover	43.0
CONTRAST	although, but, while	32.2
CAUSE-EFFECT	because, and so, thus, therefore	12.1
EQUIVALENCE	in fact, alternatively, similarly	6.0
CHANGE-TOPIC	by the way, incidentally, and now, meanwhile, well	5.1
EXAMPLE	for example, for instance	1.5
OTHER	most of all, in general	0.2

3 Identification using Lexical Information

The system has two components; one is to identify the discourse relation using lexical information, described in this section, and the other is to identify it using phrasal patterns, described in the next section.

A pair of words in two consecutive sentences can be a clue to identify the discourse relation of those sentences. For example, the CONTRAST relation may hold between two sentences which

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 133–136,
New York, June 2006. ©2006 Association for Computational Linguistics

have antonyms, such as "*ideal*" and "*reality*" in Example 1. Also, the EXAMPLE relation may hold when the second sentence has hyponyms of a word in the first sentence. For example, "*gift shop*", "*department store*", and "*supermarket*" are hyponyms of "*store*" in Example 2.

Ex1)

a. It is *ideal* that people all over the world accept independence and associate on an equal footing with each other.

b. (However,) *Reality* is not that simple.

Ex2)

a. Every town has many *stores*.

b. (For example,) *Gift shops*, *department stores*, and *supermarkets* are the main stores.

In our experiment, we used a corpus from the Web (about 20G of text) and 38 years of newspapers. We extracted pairs of sentences in which an unambiguous discourse cue phrase appears at the beginning of the second sentence. We extracted about 1,300,000 sentence pairs from the Web and about 150,000 pairs from newspapers. 300 pairs (50 of each discourse relation) were set aside as a test corpus.

3.1 Extracting Word Pairs

Word pairs are extracted from two sentences; i.e. one word from each sentence. In order to reduce noise, the words are restricted to *common nouns*, *verbal nouns*, *verbs*, and *adjectives*. Also, the word pairs are restricted to particular kinds of POS combinations in order to reduce the impact of word pairs which are not expected to be useful in discourse relation identification. We confined the combinations to the pairs involving the same part of speech and those between *verb* and *adjective*, and between *verb* and *verbal noun*.

All of the extracted word pairs are used in base form. In addition, each word is annotated with a positive or negative label. If a phrase segment includes negative words like "not", the words in the same segment are annotated with a negative label. Otherwise, words are annotated with a positive label. We don't consider double negatives. In Example 1-b, "simple" is annotated with a negative, as it includes "not" in the same segment.

3.2 Score Calculation

All possible word pairs are extracted from the sentence pairs and the frequencies of pairs are counted for each discourse relation. For a new (test) sentence pair, two types of score are calculated for each discourse relation based on all of the word pairs found in the two sentences. The scores are given by formulas (1) and (2). Here *Freq(dr, wp)* is the frequency of word pair (*wp*) in the discourse relation (*dr*). Score$_1$ is the fraction of the given discourse relation among all the word pairs in the sentences. Score$_2$ incorporates an adjustment based on the rate ($Rate_{DR}$) of the discourse relation in the corpus, i.e. the third column in Table 1. The score actually compares the ratio of a discourse relation in the particular word pairs against the ratio in the entire corpus. It helps the low frequency discourse relations get better scores.

$$Score_1(DR) = \frac{\sum_{wp} Freq(DR, wp)}{\sum_{dr,wp} Freq(dr, wp)} \qquad (1)$$

$$Score_2(DR) = \frac{\sum_{wp} Freq(DR, wp)}{\sum_{dr,wp} Freq(dr, wp) \times Rate_{DR}} \qquad (2)$$

4 Identification using Phrasal Pattern

We can sometimes identify the discourse relation between two sentences from fragments of the two sentences. For example, the CONTRAST relation is likely to hold between the pair of fragments "*… should have done ….*" and "*… did ….*", and the EXAMPLE relation is likely to hold between the pair of fragments "*There is…*" and "*Those are … and so on.*". Here "…" represents any sequence of words. The above examples indicate that the discourse relation between two sentences can be recognized using fragments of the sentences even if there are no clues based on the sort of content words involved in the word pairs. Accumulating such fragments in Japanese, we observe that these fragments actually form a phrasal pattern. A phrase (bunsetsu) in Japanese is a basic component of sentences, and consists of one or more content words and zero or more function words. We specify that a phrasal pattern contain at least three subphrases, with at least one from each sentence. Each subphrase contains the function words of the phrase, and may also include accompanying content words. We describe the method to create patterns in three steps using an example sentence pair (Example 3) which actually has the CONTRAST relation.

Ex3)

a. “kanojo-no kokoro-ni donna omoi-ga at-ta-ka-ha wakara-nai.” (No one knows what feeling she had in her mind.)

b. “sore-ha totemo yuuki-ga iru koto-dat-ta-ni-chigai-nai.” (I think that she must have needed courage.)

1) Deleting unnecessary phrases

Noun modifiers using “no” (a typical particle for a noun modifier) are excised from the sentences, as they are generally not useful to identify a discourse relation. For example, in the compound phrase “kanozyo-no (her) kokoro (mind)” in Example 3, the first phrase (her), which just modifies a noun (mind), is excised. Also, all of the phrases which modify excised phrases, and all but the last phrase in a conjunctive clause are excised.

2) Restricting phrasal pattern

In order to avoid meaningless phrases, we restrict the phrase participants to components matching the following regular expression pattern. Here, *noun-x* means all types of nouns except common nouns, i.e. verbal nouns, proper nouns, pronouns, etc.

“(*noun-x* | *verb* | *adjective*)? (*particle* | *auxiliary verb* | *period*)+$”, or “*adverb*$”

3) Combining phrases and selecting words in a phrase

All possible combinations of phrases including at least one phrase from each sentence and at least three phrases in total are extracted from a pair of sentences in order to build up phrasal patterns. For each phrase which satisfies the regular expression in 2), the subphrases to be used in phrasal patterns are selected based on the following four criteria (A to D). In each criterion, a sample of the result pattern (using all the phrases in Example 3) is expressed in bold face. Note that it is quite difficult to translate those patterns into English as many function words in Japanese are encoded as a position in English. We hope readers understand the procedure intuitively.

A) Use all components in each phrase

kanojo-no kokoro-**ni** donna omoi-**ga at-ta-ka-ha wakara-nai**.
sore-ha totemo yuuki-**ga** iru **koto-dat-ta-ni-chigai-nai**.

B) Remove *verbal noun* and *proper noun*

kanojo-no kokoro-**ni** donna omoi-**ga at-ta-ka-ha wakara-nai**.
sore-ha totemo yuuki-**ga** iru **koto-dat-ta-ni-chigai-nai**.

C) In addition, remove *verb* and *adjective*

kanojo-no kokoro-**ni** donna omoi-**ga** at-**ta-ka-ha** wakara-**nai**.
sore-ha totemo yuuki-**ga** iru **koto-dat-ta-ni-chigai-nai**.

D) In addition, remove *adverb* and remaining *noun*

kanojo-no kokoro-**ni** donna omoi-**ga** at-**ta-ka-ha** wakara-**nai**.
sore-**ha** totemo yuuki-**ga** iru koto-**dat-ta-ni-chigai-nai**.

4.1 Score Calculation

By taking combinations of 3 or more subphrases produced as described above, 348 distinct patterns can be created for the sentences in Example 3; all of them are counted with frequency 1 for the CONTRAST relation. Like the score calculation using lexical information, we count the frequency of patterns for each discourse relation over the entire corpus. Patterns appearing more than 1000 times are not used, as those are found not useful to distinguish discourse relations.

The scores are calculated replacing *Freq(dr, wp)* in formulas (1) and (2) by *Freq(dr, pp)*. Here, *pp* is a phrasal pattern and *Freq(dr, pp)* is the number of times discourse relation *dr* connects sentences for which phrasal pattern *pp* is matched. These scores will be called Score$_3$ and Score$_4$, respectively.

5 Evaluation

The system identifies one of six discourse relations, described in Table 1, for a test sentence pair. Using the 300 sentence pairs set aside earlier (50 of each discourse relation type), we ran two experiments for comparison purposes: one using only lexical information, the other using phrasal patterns as well. In the experiment using only lexical information, the system selects the relation maximizing Score$_2$ (this did better than Score$_1$). In the other, the system chooses a relation as follows: if one relation maximizes both Score$_1$ and Score$_2$, choose that relation; else, if one relation maximizes both $Score_3$ and $Score_4$, choose that relation; else choose the relation maximizing $Score_2$.

Table 2 shows the result. For all discourse relations, the results using phrasal patterns are better or the same. When we consider the frequency of discourse relations, i.e. 43% for ELABORATION, 32% for CONTRAST etc., the weighted accuracy was 53% using only lexical information, which is comparable to the similar experiment by (Marcu and Echihabi 2002) of 49.7%. Using phrasal patterns, the accuracy improves 12% to 65%. Note that the baseline accuracy (by always selecting the most frequent relation) is 43%, so the improvement is significant.

Table 2. The result

Discourse relation	Lexical info. Only	With phrasal pattern
ELABORATION	44% (22/50)	52% (26/50)
CONTRAST	62% (31/50)	86% (43/50)
CAUSE-EFFECT	56% (28/50)	56% (28/50)
EQUIVALENCE	58% (29/50)	58% (29/50)
CHANGE-TOPIC	66% (33/50)	72% (36/50)
EXAMPLE	56% (28/50)	60% (30/50)
Total	57% (171/300)	64% (192/300)
Weighted accuracy	53%	65%

Since they are more frequent in the corpus, ELABORATION and CONTRAST are more likely to be selected by $Score_1$ or $Score_3$. But adjusting the influence of rate bias using $Score_2$ and $Score_4$, it sometimes identifies the other relations.

The system makes many mistakes, but people also may not be able to identify a discourse relation just using the two sentences if the cue phrase is deleted. We asked three human subjects (two of them are not authors of this paper) to do the same task. The total (un-weighted) accuracies are 63, 54 and 48%, which are about the same or even lower than the system performance. Note that the subjects are allowed to annotate more than one relation (Actually, they did it for 3% to 30% of the data). If the correct relation is included among their *N* choices, then *1/N* is credited to the accuracy count. We measured inter annotator agreements. The average of the inter-annotator agreements is 69%. We also measured the system performance on the data where all three subjects identified the correct relation, or two of them identified the correct relation and so on (Table 3). We can see the correlation between the number of subjects who answered correctly and the system accuracy. In short, we can observe from the result and the analyses that the system works as well as a human does under the condition that only two sentences can be read.

Table 3. Accuracy for different agreements

# of subjects correct	3	2	1	0
System accuracy	71%	63%	60%	47%

.

6 Conclusion

In this paper, we proposed a system which identifies discourse relations between two successive sentences in Japanese. On top of the lexical information previously proposed, we used phrasal pattern information. Using phrasal information improves accuracy 12%, from 53% to 65%. The accuracy is comparable to human performance. There are many future directions, which include 1) applying other machine learning methods, 2) analyzing discourse relation categorization strategy, and 3) including a longer context beyond two sentences.

Acknowledgements

This research was partially supported by the National Science Foundation under Grant IIS-00325657. This paper does not necessarily reflect the position of the U.S. Government. We would like to thank Prof. Ralph Grishman, New York University, who provided useful suggestions and discussions.

References

Daniel Marcu and Abdessamad Echihabi. 2002. An Unsupervised Approach to Recognizing Discourse Relations, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* 368-375.

Florian Wolf and Edward Gibson. 2005. Representing Discourse Coherence: A Corpus-Based Study, *Computational Linguistics*, 31(2):249-287.

Takashi Ichikawa. 1978. Syntactic Overview for Japanese Education, Kyo-iku publishing, 65-67 (in Japanese).

Weblog Classification for Fast Splog Filtering: A URL Language Model Segmentation Approach

Franco Salvetti[†⋆]
franco.salvetti@colorado.edu

Nicolas Nicolov[⋆]
nicolas@umbrialistens.com

[†]Dept. of Computer Science, Univ. of Colorado at Boulder, 430 UCB, Boulder, CO 80309-0430
[⋆]Umbria, Inc., 1655 Walnut Str, Boulder, CO 80302

Abstract

This paper shows that in the context of statistical weblog classification for splog filtering based on n-grams of tokens in the URL, further segmenting the URLs beyond the standard punctuation is helpful. Many splog URLs contain phrases in which the words are glued together in order to avoid splog filtering techniques based on punctuation segmentation and unigrams. A technique which segments long tokens into the words forming the phrase is proposed and evaluated. The resulting tokens are used as features for a weblog classifier whose accuracy is similar to that of humans (78% vs. 76%) and reaches 93.3% of precision in identifying splogs with recall of 50.9%.

1 Introduction

The blogosphere, which is a subset of the web and is comprised of personal electronic journals (weblogs) currently encompasses 27.2 million pages and doubles in size every 5.5 months (Technorati, 2006). The information contained in the blogosphere has been proven valuable for applications such as marketing intelligence, trend discovery, and opinion tracking (Hurst, 2005). Unfortunately in the last year the blogosphere has been heavily polluted with spam weblogs (called *splogs*) which are weblogs used for different purposes, including promoting affiliated websites (Wikipedia, 2006). Splogs can skew the results of applications meant to quantitatively analyze the blogosphere. Sophisticated content-based methods or methods based on link analysis (Gyöngyi et al., 2004), while providing effective splog filtering, require extra web crawling and can be slow. While a combination of approaches is necessary to provide adequate splog filtering, similar to (Kan & Thi, 2005), we propose, as a preliminary step in the overall splog filtering, a fast, lightweight and accurate method merely based on the analysis of the URL of the weblog without considering its content.

For quantitative and qualitative analysis of the content of the blogosphere, it is acceptable to eliminate a small fraction of good data from analysis as long as the remainder of the data is splog-free. This elimination should be kept to a minimum to preserve counts needed for reliable analysis. When using an ensemble of methods for comprehensive splog filtering it is acceptable for pre-filtering approaches to lower recall in order to improve precision allowing more expensive techniques to be applied on a smaller set of weblogs. The proposed method reaches 93.3% of precision in classifying a weblog in terms of `spam` or `good` if 49.1% of the data are left aside (labeled as `unknown`). If all data needs to be classified our method achieves 78% accuracy which is comparable to the average accuracy of humans (76%) on the same classification task.

Sploggers, in creating splogs, aim to increase the traffic to specific websites. To do so, they frequently communicate a concept (e.g., a service or a product) through a short, sometimes non-grammatical phrase embedded in the URL of the weblog (e.g., `http://adult-video-mpegs.blogspot.com`). We want to build a statistical classifier which leverages the language used in these descriptive URLs in order to classify weblogs as `spam` or `good`. We built an initial language model-based classifier on the tokens of the URLs after tokenizing on punctuation (`.`, `-`,

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 137–140,
New York, June 2006. ©2006 Association for Computational Linguistics

_, /, ?, =, etc.). We ran the system and got an accuracy of 72.2% which is close to the accuracy of humans—76% (the baseline is 50% as the training data is balanced). When we did error analysis on the misclassified examples we observed that many of the mistakes were on URLs that contain words glued together as one token (e.g., `dailyfreeipod`). Had the words in these tokens been segmented the initial system would have classified the URL correctly. We, thus, turned our attention to additional segmenting of the URLs beyond just punctuation and using this intra-token segmentation in the classification.

Training a segmenter on standard available text collections (e.g., PTB or BNC) did not seem the way to procede because the lexical items used and the sequence in which they appear differ from the usage in the URLs. Given that we are interested in unsupervised lightweight approaches for URL segmentation, one possibility is to use the URLs themselves after segmenting on punctuation and to try to learn the segmenting (the majority of URLs are naturally segmented using punctuation as we shall see later). We trained a segmenter on the tokens in the URLs, unfortunately this method did not provide sufficient improvement over the system which uses tokenization on punctuation. We hypothesized that the content of the splog pages corresponding to the splog URLs could be used as a corpus to learn the segmentation. We crawled 20K weblogs corresponding to the 20K URLs labeled as `spam` and `good` in the training set, converted them to text, tokenized and used the token sequences as training data for the segmenter. This led to a statistically significant improvement of 5.8% of the accuracy of the splog filter.

2 Engineering of splogs

Frequently sploggers indicate the semantic content of the weblogs using descriptive phrases—often noun groups (non-recursive noun phrases) like `adult-video-mpegs`. There are different varieties of splogs: commercial products (especially electronics), vacations, mortgages, and adult-related.

Users don't want to see splogs in their results and marketing intelligence applications are affected when data contains splogs. Existing approaches to splog filtering employ statistical classifiers (e.g., SVMs) trained on the tokens in a URL after tokenization on punctuation (Kolari et al., 2006). To avoid being identified as a splog by such systems one of the creative techniques that sploggers use is to glue words together into longer tokens for which there will not be statistical information (e.g., `businessopportunitymoneyworkathome` is unlikely to appear in the training data while `business`, `opportunity`, `money`, `work`, `at` and `home` are likely to have been seen in training). Another approach to dealing with splogs is having a list of splog websites (SURBL, 2006). Such an approach based on blacklists is now less effective because bloghosts provide tools which can be used for the automatic creation of a large quantity of splogs.

3 Splog filtering

The weblog classifier uses a segmenter which splits the URL in tokens and then the token sequence is used for supervised learning and classification.

3.1 URL segmentation

The segmenter first tokenizes the URLs on punctuation symbols. Then the current URL tokens are examined for further possible segmentation. The segmenter uses a sliding window of n (e.g., 6) characters. Going from left to right in a greedy fashion the segmenter decides whether to split after the current third character. Figure 1 illustrates the processing of `www.dietthatworks.com` when considering the token `dietthatworks`. The character '`o`' indicates that the left and right tri-grams are kept together while '•' indicates a point where the segmenter decides a break should occur. The segmentation decisions are based on counts collected during training. For example, during the segmentation of `dietthatworks` in the case of `i e t` • `t h a` we essentially consider how many times we have seen in the training data the 6-gram '`iettha`' vs. '`iet␣tha`'. Certain characters (e.g., digits) are generalized both during training and segmentation.

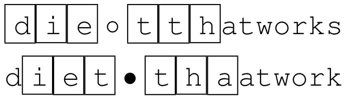


Figure 1: Workings of the segmenter

3.2 Classification

For the weblog classification a simple Naïve Bayes classifier is used. Given a token sequence $T = \langle t_1, \ldots, t_n \rangle$, representing the segmented URL, the class $\hat{c} \in C = \{\texttt{spam}, \texttt{good}\}$ is decided as:

$$\begin{aligned}
\hat{c} &= \arg\max_{c \in C} P(c|T) = \arg\max_{c \in C} \frac{P(c) \cdot P(T|c)}{P(T)} \\
&= \arg\max_{c \in C} P(c) \cdot P(T|c) \\
&= \arg\max_{c \in C} P(c) \cdot \prod_{i=1}^{n} P(t_i|c)
\end{aligned}$$

In the last step we made the conditional independence assumption. For calculating $P(t_i|c)$ we use Laplace (add one) smoothing (Jurafsky & Martin, 2000). We have also explored classification via simple voting techniques such as:

$$a = sgn \sum_{i=1}^{n} sgn \left(P(t_i|\texttt{spam}) - P(t_i|\texttt{good}) \right)$$

$$\hat{c} = \begin{cases} \texttt{spam}, & if \quad a = 1 \\ \texttt{good}, & otherwise \end{cases}$$

Because we are interested in having control over the precision/recall of the classifier we introduce a score meant to be used for deciding whether to label a URL as unknown.

$$score(T) = \left| \frac{P(\texttt{spam}|T) - P(\texttt{good}|T)}{P(\texttt{spam}|T) + P(\texttt{good}|T)} \right|$$

If $score(T)$ exceeds a certain threshold τ we label T as spam or good using the greater probability of $P(\texttt{spam}|T)$ or $P(\texttt{good}|T)$. To control the presicion of the classifier we can tune τ. For instance, when we set $\tau = 0.75$ we achieve 93.3% of precision which implied a recall of 50.9%. An alternate commonly used technique to compute a score is to look at the log likelihood ratio.

4 Experiments and results

First we discuss the segmenter. 10,000 spam and 10,000 good weblog URLs and their corresponding HTML pages were used for the experiments. The 20,000 weblog HTML pages are used to induce the segmenter. The first experiment was aimed at finding how common extra segmentation beyond punctuation is as a phenomenon. The segmenter was run on the actual training URLs. The number of URLs that are additionally segmented besides the segmentation on punctuation are reported in Table 1.

# of splits	# spam URLs	# good URLs
1	2,235	2,274
2	868	459
3	223	46
4	77	7
5	2	1
6	4	1
8	3	–
Total	3,412	2,788

Table 1: Number of extra segmentations in a URL

The multiple segmentations need not all occur on the same token in the URL after initial segmentation on punctuations.

The segmenter was then evaluated on a separate test set of 1,000 URLs for which the ground truth for the segmentation was marked. The results are in Table 2. The evaluation is only on segmentation events and does not include tokenization decisions around punctuation.

Precision	Recall	F-measure
84.31	48.84	61.85

Table 2: Performance of the segmenter

Figure 2 shows long tokens which are correctly split. The weblog classifier was then run on the test set. The results are shown in Table 3.

cash • for • your • house
unlimitted • pet • supllies
jim • and • body • fat
weight • loss • product • info
kick • the • boy • and • run
bringing • back • the • past
food • for • your • speakers

Figure 2: Correct segmentations

accuracy	78%
prec. spam	82%
rec. spam	71%
f-meas spam	76%
prec. good	74%
rec. good	84%
f-meas good	79%

Table 3: Classification results

The performance of humans on this task was also evaluated. Eight individuals performed the splog identification just looking at the unsegmented URLs. The results for the human annotators are given in Table 4. The average accuracy of the humans (76%) is similar to that of the system (78%).

	Mean	σ
accuracy	76%	6.71
prec. spam	83%	7.57
rec. spam	65%	6.35
f-meas spam	73%	7.57
prec. good	71%	6.35
rec. good	87%	6.39
f-meas good	78%	6.08

Table 4: Results for the human annotators

From an information retrieval perspective if only 50.9% of the URLs are retrieved (labelled as either `spam` or `good` and the rest are labelled as `unknown`) then of the `spam`/`good` decisions 93.3% are correct. This is relevant for cases where a URL splog filter is in cascade followed by, for example, a content-based one.

5 Discussion

The system performs better with the intra-token segmentation because the system is forced to guess unseen events on fewer occasions. For instance given the input URL `www.ipodipodipod.com` in the system which segments solely on punctuation both the `spam` and the `good` model will have to guess the probability of `ipodipodipod` and the results depend merely on the smoothing technique.

Even if we reached the average accuracy of humans we expect to be able to improve the system further as the maximum accuracy among the human annotators is 90%. Among the errors of the segmenter the most common are related to plural nouns (‘`girl•s`’ vs. ‘`girls`’) and past tense of verbs (‘`dedicate•d`’ vs. ‘`dedicated`’).
The proposed approach has ramifications for splog filtering systems that want to consider the outward links from a weblog.

6 Conclusions

We have presented a technique for determining whether a weblog is splog based merely on alalyzing its URL. We proposed an approach where we initially segment the URL in words and then do the classification. The technique is simple, yet very effective—our system reaches an accuracy of 78% (while humans perform at 76%) and 93.3% of precision in classifying a weblog with recall of 50.9%.

Acknowledgements. We wish to thank Ted Kremer, Howard Kaushansky, Ash Beits, Allen Bennett, Susanne Costello, Hillary Gustave, Glenn Meuth, Micahel Sevilla and Ron Woodward for help with the experiments and comments on an earlier draft.

References

Gyöngyi, Zoltan, Hector Garcia-Molina & Jan Pedersen. 2004. “Combating Web Spam with TrustRank”. *Proceedings of the 30th International Conference on Very Large Data Bases* (*VLDB*).

Matthew Hurst. 2005. “Deriving Marketing Intelligence from Online Discussion”. *11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining* (*KDD05*), 419-428. Chicago, Illinois, USA.

Jurafsky, D. & J.H. Martin. 2000. *Speech and Language Processing*. Upper Saddle River, NJ: Prentice Hall.

Min-Yen Kan & Hoang Oanh Nguyen Thi. 2005. “Fast Webpage Classification Using URL Features”. *14th ACM international conference on Information and Knowledge Management*, 325-326.

Kolari, Pranam, Tim Finin & Anupam Joshi. 2006. “SVMs for the Blogosphere: Blog Identification and Splog Detection”. *AAAI Symposium on Computational Approaches to Analyzing Weblogs*, 92-99. Stanford.

SURBL. 2006. *SURBL — Spam URI Realtime Blocklists*, `http://www.surbl.org`

Technorati. 2006. *State of the Blogosphere, February 2006 Part 1: On Blogosphere Growth*, `technorati.com/weblog/2006/02/81.html`

Wikipedia. 2006. *Splog (Spam blog)*, `http://en.wikipedia.org/wiki/Splog`

Word Domain Disambiguation via Word Sense Disambiguation

Antonio Sanfilippo, Stephen Tratz, Michelle Gregory
Pacific Northwest National Laboratory
Richland, WA 99352
{Antonio.Sanfilippo, Stephen.Tratz, Michelle.Gregory}@pnl.gov

Abstract

Word subject domains have been widely used to improve the performance of word sense disambiguation algorithms. However, comparatively little effort has been devoted so far to the disambiguation of word subject domains. The few existing approaches have focused on the development of algorithms specific to word domain disambiguation. In this paper we explore an alternative approach where word domain disambiguation is achieved via word sense disambiguation. Our study shows that this approach yields very strong results, suggesting that word domain disambiguation can be addressed in terms of word sense disambiguation with no need for special purpose algorithms.

1 Introduction

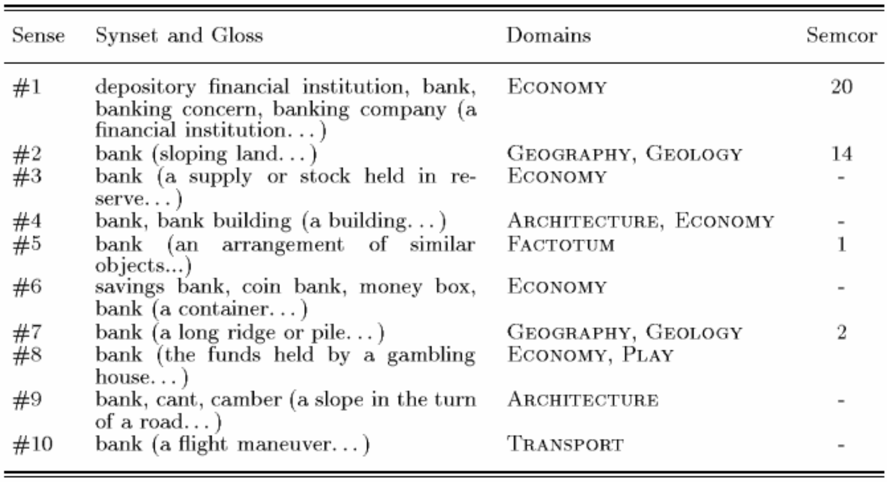
Word subject domains have been ubiquitously used in dictionaries to help human readers pinpoint the specific sense of a word by specifying technical usage, e.g. see “subject field codes” in Procter (1978). In computational linguistics, word subject domains have been widely used to improve the performance of machine translation systems. For example, in a review of commonly used features in automated translation, Mowatt (1999) reports that most of the machine translation systems surveyed made use of word subject domains. Word subject domains have also been used in information systems. For example, Sanfilippo (1998) describes a summarization system where subject domains provide users with useful conceptual parameters to tailor summary requests to a user’s interest.

Successful usage of word domains in applications such as machine translation and summarization is strongly dependent on the ability to assign the appropriate subject domain to a word in its context. Such an assignment requires a process of Word Domain Disambiguation (WDD) because the same word can often be assigned different subject domains out of context (e.g. the word `partner` can potentially be related to FINANCE or MARRIAGE).

Interestingly enough, word subject domains have been widely used to improve the performance of Word Sense Disambiguation (WSD) algorithms (Wilks and Stevenson 1998, Magnini et al. 2001; Gliozzo et al. 2004). However, comparatively little effort has been devoted so far to the word domain disambiguation itself. The most notable exceptions are the work of Magnini and Strapparava (2000) and Suarez & Palomar (2002). Both studies propose algorithms specific to the WDD task and have focused on the disambiguation of noun domains.

In this paper we explore an alternative approach where word domain disambiguation is achieved via word sense disambiguation. Moreover, we extend the treatment of WDD to verbs and adjectives. Initial results show that this approach yield very strong results, suggesting that WDD can be addressed in terms of word sense disambiguation with no need of special purpose algorithms.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 141–144,
New York, June 2006. ©2006 Association for Computational Linguistics



Sense	Synset and Gloss	Domains	Semcor
#1	depository financial institution, bank, banking concern, banking company (a financial institution. . .)	ECONOMY	20
#2	bank (sloping land. . .)	GEOGRAPHY, GEOLOGY	14
#3	bank (a supply or stock held in reserve. . .)	ECONOMY	-
#4	bank, bank building (a building. . .)	ARCHITECTURE, ECONOMY	-
#5	bank (an arrangement of similar objects...)	FACTOTUM	1
#6	savings bank, coin bank, money box, bank (a container. . .)	ECONOMY	-
#7	bank (a long ridge or pile. . .)	GEOGRAPHY, GEOLOGY	2
#8	bank (the funds held by a gambling house. . .)	ECONOMY, PLAY	
#9	bank, cant, camber (a slope in the turn of a road. . .)	ARCHITECTURE	-
#10	bank (a flight maneuver. . .)	TRANSPORT	-

Figure 1: Senses and domains for the word *bank* in WordNet Domains, with number of occurrences in SemCor, adapted from Magnini et al. (2002).

2 WDD via WSD

Our approach relies on the use of WordNet Domains (Bagnini and Cavaglià 2000) and can be outlined in the following two steps:

1. use a WordNet-based WSD algorithm to assign a sense to each word in the input text, e.g. `doctor` → `doctor#n#1`
2. use WordNet Domains to map disambiguated words into the subject domain associated with the word, e.g. `doctor#n#1`→`doctor#n#1#`MEDICINE.

2.1 WordNet Domains

WordNet Domains is an extension of WordNet (http://wordnet.princeton.edu/) where synonym sets have been annotated with one or more subject domain labels, as shown in Figure 1. Subject domains provide an interesting and useful classification which cuts across part of speech and WordNet sub-hierarchies. For example, `doctor#n#1` and `operate#n#1` both have subject domain MEDICINE, and SPORT includes both `athlete#n#1` with top hypernym `lifeform#n#1` and `sport#n#1` with top hypernym `act#n#2`.

2.2 Word Sense Disambiguation

To assign a sense to each word in the input text, we used the WSD algorithm presented in Sanfilippo et al. (2006). This WSD algorithm is based on a supervised classification approach that uses SemCor[1] as training corpus. The algorithm employs the OpenNLP MaxEnt implementation of the maximum entropy classification algorithm (Berger et al. 1996) to develop word sense recognition signatures for each lemma which predicts the most likely sense for the lemma according to the context in which the lemma occurs.

Following Dang & Palmer (2005) and Kohomban & Lee (2005), Sanfilippo et al. (2006) use contextual, syntactic and semantic information to inform our verb class disambiguation system.

- Contextual information includes the verb under analysis plus three tokens found on each side of the verb, within sentence boundaries. Tokens included word as well as punctuation.
- Syntactic information includes grammatical dependencies (e.g. subject, object) and morpho-syntactic features such as part of speech, case, number and tense.
- Semantic information includes named entity types (e.g. person, location, organization) and hypernyms.

We chose this WSD algorithm as it provides some of the best published results to date, as the comparison with top performing WSD systems in Senseval3 presented in Table 1 shows---see http://www.senseval.org and Snyder & Palmer (2004) for terms of reference on Senseval3.

[1] http://www.cs.unt.edu/~rada/downloads.html.

System	Precision	Fraction of Recall
Sanfilippo et al. 2006	61%	22%
GAMBL	59.0%	21.3%
SenseLearner	56.1%	20.2%
Baseline	52.9%	19.1%

Table 1: Results for verb sense disambiguation on Senseval3 data, adapted from Sanfilippo et al. (2006).

3 Evaluation

To evaluate our WDD approach, we used both the SemCor and Senseval3 data sets. Both corpora were stripped of their sense annotations and processed with an extension of the WSD algorithm of Sanfilippo et al. (2006) to assign a WordNet sense to each noun, verb and adjective. The extension consisted in extending the training data set so as to include a selection of WordNet examples (full sentences containing a main verb) and the Open Mind Word Expert corpus (Chklovski and Mihalcea 2002).

The original hand-coded word sense annotations of the SemCor and Senseval3 corpora and the word sense annotations assigned by the WSD algorithm used in this study were mapped into subject domain annotations using WordNet Domains, as described in the opening paragraph of section 2 above. The version of the SemCor and Senseval3 corpora where subject domain annotations were generated from hand-coded word senses served as gold standard. A baseline for both corpora was obtained by assigning to each lemma the subject domain corresponding to sense 1 of the lemma.

WDD results of a tenfold cross-validation for the SemCor data set are given in Table 2. Accuracy is high across nouns, verbs and adjectives.[2] To verify the statistical significance of these results against the baseline, we used a standard proportions comparison test (see Fleiss 1981, p. 30). According to this test, the accuracy of our system is significantly better than the baseline.

The high accuracy of our WDD algorithm is corroborated by the results for the Senseval3 data set in Table 3. Such corroboration is important as the Senseval3 corpus was not part of the data set used to train the WSD algorithm which provided the basis for subject domain assignment. The standard comparison test for the Senseval3 is not as conclusive as with SemCor. This is probably due to the comparatively smaller size of the Senseval3 corpus.

	Nouns	Verbs	Adj.s	Overall
Accuracy	0.874	0.933	0.942	0.912
Baseline	0.848	0.927	0.932	0.897
p-value	4.6e-54	1.4e-07	5.5e-08	1.4e-58

Table 2: SemCor WDD results.

	Nouns	Verbs	Adj.s	Overall
Accuracy	0.797	0.908	0.888	0.848
Baseline	0.783	0.893	0.862	0.829
p-value	0.227	0.169	0.151	0.048

Table 3: Senseval3 WDD results.

4 Comparison with Previous WDD Work

Our WDD algorithm compares favorably with the approach explored in Bagnini and Strapparava (2000), who report 0.82 p/r in the WDD tasks for a subset of nouns in SemCor.

Suarez and Palomar (2002) report WDD results of 78.7% accuracy for nouns against a baseline of 68.7% accuracy for the same data set. As in the present study, Suarez and Palomar derive the baseline by assigning to each lemma the subject domain corresponding to sense 1 of the lemma. Unfortunately, a meaningful comparison with Suarez and Palomar (2002) is not possible as they use a different data set, the DSO corpus.[3] We are currently working on repeating our study with the DSO corpus and will include the results of this evaluation in the final version of the paper to achieve commensurability with the results reported by Suarez and Palomar.

5 Conclusions and Further Work

Current approaches to WDD have assumed that special purpose algorithms are needed to model the WDD task. We have shown that very competitive and perhaps unrivaled results (pending on evaluation of our WDD algorithm with the DSO corpus) can be obtained using WSD as the basis for subject domain assignment. This improvement in WDD performance can be used to

[2] We have not worked on adverbs yet, but we expect comparable results.

[3] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC97T12.

obtain further gains in WSD accuracy, following Wilks and Stevenson (1998), Magnini et al. (2001) and Gliozzo et al. (2004). A more accurate WSD model will in turn yield yet better WDD results, as demonstrated in this paper. Consequently, further improvements in accuracy for both WSD and WDD can be expected through a bootstrapping cycle where WDD results are fed as input to the WSD process, and the resulting improved WSD model is then used to achieve better WDD results. We intend to explore this possibility in future extensions of this work.

Acknowledgements

We would like to thank Paul Whitney for help with the evaluation of the results presented in Section 3.

References

Berger, A., S. Della Pietra and V. Della Pietra (1996) A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics,* volume 22, number 1, pages 39-71.

Chklovski, T. and R. Mihalcea (2002) Building a Sense Tagged Corpus with Open Mind Word Expert. Proceedings of the ACL 2002 Workshop on "Word Sense Disambiguation: Recent Successes and Future Directions, Philadelphia, July 2002, pp. 116-122.

Dang, H. T. and M. Palmer (2005) The Role of Semantic Roles in Disambiguating Verb Senses. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor MI, June 26-28, 2005.

Fleiss, J. L. (1981) *Statistical Methods for Rates and Proportions*. 2nd edition. New York: John Wiley & Sons.

Gliozzo, A., C. Strapparava, I. Dagan (2004) Unsupervised and Supervised Exploitation of Semantic Domains in Lexical Disambiguation. *Computer Speech and Language*,18(3), Pages 275-299.

Kohomban, U. and W. Lee (2005) Learning semantic classes for word sense disambiguation. In *Proceedings of the 43rd Annual meeting of the Association for Computational Linguistics*, Ann Arbor, MI.

Magnini, B., Cavaglià, G. (2000) Integrating Subject Field Codes into WordNet. *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, Athens, Greece, 31 MAY- 2 JUNE 2000, pp. 1413-1418.

Magnini, B., Strapparava C. (2000) Experiments in Word Domain Disambiguation for Parallel Texts. *Proceedings of the ACL Workshop on Word Senses and Multilinguality*, Hong-Kong, October 7, 2000, pp. 27-33

Magnini, B., C. Strapparava, G. Pezzulo and A. Gliozzo (2001) Using Domain Information for Word Sense Disambiguation. In Proceeding of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems, pp. 111-114, 5-6 July 2001, Toulouse, France.

Magnini, B., C. Strapparava, G. Pezzulo and A. Gliozzo (2002) The Role of Domain Information in Word Sense Disambiguation. *Natural Language Engineering*, 8(4):359—373.

Mowatt, D. (1999) Types of Semantic Information Necessary in a Machine Translation Lexicon. *Conférence TALN*, Cargèse, pp. 12-17.

Procter, Paul (Ed.) (1978) *Longman Dictionary o Contemporary English*. Longman Group Ltd., Essex, UK.

Sanfilippo, A. (1998) Ranking Text Units According to Textual Saliency, Connectivity and Topic Aptness. *COLING-ACL 1998*: 1157-1163.

Sanfilippo, A., S. Tratz, M. Gregory, A.Chappell, P. Whitney, C. Posse, P. Paulson, B. Baddeley, R. Hohimer, A. White. (2006) Automating Ontological Annotation with WordNet. *Proceedings of the 3rd Global WordNet Conference*, Jeju Island, South Korea, Jan 19-26 2006.

Snyder, B. and M. Palmer. 2004. The English all-words task. *SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain.

Suárez, A., Palomar, M. (2002) Word sense vs. word domain disambiguation: a maximum entropy approach. In Sojka P., Kopecek I., Pala K., eds.: Text, Speech and Dialogue (TSD 2002). Volume 2448 of Lecture Notes in Artificial Intelligence, Springer, (2002) 131—138.

Wilks, Y. and Stevenson, M. (1998) Word sense disambiguation using optimised combinations of knowledge sources. *Proceedings of the 17th international conference on Computational Linguistics*, pp. 1398—1402.

Selecting relevant text subsets from web-data for building topic specific language models

Abhinav Sethy, Panayiotis G. Georgiou, Shrikanth Narayanan
Speech Analysis and Interpretation Lab
Integrated Media Systems Center
Viterbi School of Engineering
Department of Electrical Engineering-Systems
University of Southern California

Abstract

In this paper we present a scheme to select *relevant subsets* of sentences from a large generic corpus such as text acquired from the web. A relative entropy (R.E) based criterion is used to incrementally select sentences whose distribution matches the domain of interest. Experimental results show that by using the proposed subset selection scheme we can get significant performance improvement in both Word Error Rate (WER) and Perplexity (PPL) over the models built from the entire web-corpus by using just 10% of the data. In addition incremental data selection enables us to achieve significant reduction in the vocabulary size as well as number of n-grams in the adapted language model. To demonstrate the gains from our method we provide a comparative analysis with a number of methods proposed in recent language modeling literature for cleaning up text.

1 Introduction

One of the main challenges in the rapid deployment of NLP applications is the lack of in-domain data required for training statistical models. Language models, especially n-gram based, are key components of most NLP applications, such as speech recognition and machine translation, where they serve as priors in the decoding process. To estimate a n-gram language model we require examples of in-domain transcribed utterances, which in absence of readily available relevant corpora have to be collected manually. This poses severe constraints in terms of both system turnaround time and cost.

This led to a growing interest in using the World Wide Web (WWW) as a corpus for NLP (Lapata, 2005; Resnik and Smith, 2003). The web can serve as a good resource for automatically gathering data for building task-specific language models. Webpages of interest can be identified by generating query terms either manually or automatically from an initial set of in-domain sentences by measures such as TFIDF or Relative Entropy (R.E). These webpages can then be converted to a text corpus (which we will refer to as *web-data*) by appropriate preprocessing. However text gathered from the web will rarely fit the demands or the nature of the domain of interest completely. Even with the best queries and web crawling schemes, both the style and content of the web-data will usually differ significantly from the specific needs. For example, a speech recognition system requires conversational style text whereas most of the data on the web is literary.

The mismatch between in-domain data and web-data can be seen as a semi-supervised learning problem. We can model the web-data as a mix of sentences from two classes: in-domain (**I**) and noise (**N**) (or out-of-domain). The labels **I** and **N** are latent and unknown for the sentences in web-data but we usually have a small number of examples of in-domain examples **I**. Selecting the right labels for the unlabeled set is important for benefiting from it.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 145–148,
New York, June 2006. ©2006 Association for Computational Linguistics

Recent research on semi-supervised learning shows that in many cases (Nigam et al., 2000; Zhu, 2005) poor preprocessing of unlabeled data might actually lower the performance of classifiers. We found similar results in our language modeling experiments where the presence of a large set of noisy **N** examples in training actually lowers the performance slightly in both perplexity and WER terms. Recent literature on building language models from text acquired from the web addresses this issue partly by using various rank-and-select schemes for identifying the set **I** (Ostendorf et al., 2005; Sethy, 2005; Sarikaya, 2005). However we believe that similar to the question of balance (Zhu, 2005) in semi-supervised learning for classification, we need to address the question of distributional similarity while selecting the appropriate utterances for building a language model from noisy data. The subset of sentences from web-data which are selected to build the adaptation language should have a distribution similar to the in-domain data model.

To address the issue of distributional similarity we present an incremental algorithm which compares the distribution of the selected set and the in-domain examples by using a relative entropy (R.E) criterion. We will review in section 2 some of the ranking schemes which provide baselines for performance comparison and in section 3 we describe the proposed algorithm. Experimental results are provided in section 4, before we conclude with a summary of this work and directions for the future.

2 Rank and select methods for text cleaning

The central idea behind text cleanup schemes in recent literature, on using web-data for language modeling, is to use a scoring function that measures the similarity of each observed sentence in the web-data to the in-domain set and assigns an appropriate score. The subsequent step is to set a threshold in terms of either the minimum score or the number of top scoring sentences. The threshold can usually be fixed using a heldout set. Ostendorf (2005) use perplexity from an in-domain n-gram language model as a scoring function. More recently, a modified version of the BLEU metric which measures sentence similarity in machine translation has been proposed by Sarikaya (2005) as a scoring function. Instead of explicit ranking and thresholding it is also possible to design a classifier in a learning from positive and unlabeled examples framework (LPU) (Liu et al., 2003). In this system, a subset of the unlabeled set is selected as the negative or the noise set **N**. A two class classifier is then trained using the in-domain set and the negative set. The classifier is then used to label the sentences in the web-data. The classifier can then be iteratively refined by using a better and larger subset of the **I/N** sentences selected in each iteration.

Rank ordering schemes do not address the issue of distributional similarity and select many sentences which already have a high probability in the in-domain text. Adapting models on such data has the tendency to skew the distribution even further towards the center. For example, in our doctor-patient interaction task short sentences containing the word ‘okay’ such as ‘okay’,‘yes okay’, ‘okay okay’ were very frequent in the in-domain data. Perplexity or other similarity measures give a high score to all such examples in the web-data boosting the probability of these words even further while other pertinent sentences unseen in the in-domain data such as ‘Can you stand up please?’ are ranked low and get rejected.

3 Incremental Selection

To address the issue of distributional similarity we developed an incremental greedy selection scheme based on relative entropy which selects a sentence if adding it to the already selected set of sentences reduces the relative entropy with respect to the in-domain data distribution.

Let us denote the language model built from in-domain data as P and let P_{init} be a language model for initialization purposes which we estimate by bagging samples from the same in-domain data. To describe our algorithm we will employ the paradigm of unigram probabilities though the method generalizes to higher n-grams also.

Let $W(i)$ be a initial set of counts for the words i in the vocabulary V initialized using P_{init}. We denote the count of word i in the j^{th} sentence s_j of web-data with m_{ij}. Let $n_j = \sum_i m_{ij}$ be the number of words in the sentence and $N = \sum_i W(i)$ be the total number of words already selected. The relative entropy of the maximum likelihood estimate of the language model of the selected sentences to the initial model P is given by

$$H(j-1) = -\sum_i P(i) \ln \frac{P(i)}{W(i)/N}$$

If we select the sentence s_j , the updated R.E

$$H(j) = -\sum_i P(i) \ln \frac{P(i)}{(W(i)+m_{ij})/(N+n_j)}$$

Direct computation of R.E using the above expressions for every sentence in the web-data will have a very high computational cost since $O(V)$ computations per sentence in the web-data are required. However given the fact that m_{ij} is sparse, we can split the summation $H(j)$ into

$$\begin{aligned} H(j) &= -\sum_i P(i) \ln P(i) + \\ &\quad + \sum_i P(i) \ln \frac{W(i)+m_{ij}}{N+n_j} \\ &= H(j-1) + \underbrace{\ln \frac{N+n_j}{N}}_{T1} \\ &\quad \underbrace{- \sum_{i, m_{ij} \neq 0} P(i) \ln \frac{(W(i)+m_{ij})}{W(i)}}_{T2} \end{aligned}$$

Intuitively, the term $T1$ measures the decrease in probability mass because of adding n_j words more to the corpus and the term $T2$ measures the in-domain distribution P weighted improvement in probability for words with non-zero m_{ij}.

For the R.E to decrease with selection of sentence s_j we require $T1 < T2$. To make the selection more refined we can impose a condition $T1 + thr(j) < T2$ where $thr(j)$ is a function of j. A good choice for $thr(j)$ based on empirical study is a function that declines at the same rate as the ratio $\ln \frac{(N+n_j)}{N} \approx n_j/N \approx 1/kj$ where k is the average number of words for every sentence.

The proposed algorithm is sequential and greedy in nature and can benefit from randomization of the order in which it scans the corpus. We generate permutes of the corpus by scanning through the corpus and randomly swapping sentences. Next we do sequential selection on each permutation and merge the selected sets.

The choice of using maximum likelihood estimation for estimating the intermediate language models for $W(j)$ is motivated by the simplification in the entropy calculation which reduces the order from $O(V)$ to $O(k)$. However, maximum likelihood estimation of language models is poor when compared to smoothing based estimation. To balance the computation cost and estimation accuracy, we modify the counts $W(j)$ using Kneser-Ney smoothing periodically after fixed number of sentences.

4 Experiments

Our experiments were conducted on medical domain data collected for building the English ASR of our English-Persian Speech to Speech translation project (Georgiou et al., 2003). We have 50K in-domain sentences for this task available. We downloaded around 60GB data from the web using automatically generated queries which after filtering and normalization amount to 150M words. The test set for perplexity evaluations consists of 5000 sentences(35K words) and the heldout set had 2000 sentences (12K words). The test set for word error rate evaluation consisted of 520 utterances. A generic conversational speech language model was built from the WSJ, Fisher and SWB corpora interpolated with the CMU LM. All language models built from web-data and in-domain data were interpolated with this language model with the interpolation weight determined on the heldout set.

We first compare our proposed algorithm against baselines based on perplexity(PPL), BLEU and LPU classification in terms of test set perplexity. As the comparison shows the proposed algorithm outperforms the rank-and-select schemes with just 1/10th of data. Table 1 shows the test set perplexity with different amounts of initial in-domain data. Table 2 shows the number of sentences selected for the best perplexity on the heldout set by the above schemes. The average relative perplexity reduction is around 6%. In addition to the PPL and WER improvements we were able to acheive a factor of 5 reduction in the number of estimated language model parameters (bigram+trigram) and a 30% reduction in the vocab-

	10K	20K	30K	40K
No Web	60	49.6	42.2	39.7
AllWeb	57.1	48.1	41.8	38.2
PPL	56.1	48.1	41.8	38.2
BLEU	56.3	48.2	42.0	38.3
LPU	56.3	48.2	42.0	38.3
Proposed	**54.8**	**46.8**	**40.7**	**38.1**

Table 1: Perplexity of testdata with the web adapted model for different number of initial sentences.

ulary size. *No Web* refers to the language model built from just in-domain data with no web-data. *AllWeb* refers to the case where the entire web-data was used.

The WER results in Table 3 show that adding data from the web without proper filtering can actually harm the performance of the speech recognition system when the initial in-domain data size increases. This can be attributed to the large increase in vocabulary size which increases the acoustic decoder perplexity. The average reduction in WER using the proposed scheme is close to 3% relative. It is interesting to note that for our data selection scheme the perplexity improvments correlate surprisingly well with WER improvments. A plausible explanation is that the perplexity improvments are accompanied by a significant reduction in the number of language model parameters.

5 Conclusion and Future Work

In this paper we have presented a computationally efficient scheme for selecting a subset of data from an unclean generic corpus such as data acquired from the web. Our results indicate that with this scheme, we can identify small subsets of sentences (about 1/10th of the original corpus), with which we can build language models which are substantially smaller in size and yet have better performance in both perplexity and WER terms compared to models built using the entire corpus. Although our focus in the paper was on web-data, we believe the proposed method can be used for adaptation of topic specific models from large generic corpora.

We are currently exploring ways to use multiple bagged in-domain language models for the selection process. Instead of sequential scan of the corpus, we are exploring the use of rank-and-select methods to give a better search sequence.

	10K	20K	30K	40K
PPL	93	92	91	91
BLEU	91	90	89	89
LPU	90	88	87	87
Proposed	**12**	**11**	**11**	**12**

Table 2: Percentage of web-data selected for different number of initial sentences.

	10K	20K	30K	40K
No Web	19.8	18.9	18.3	17.9
AllWeb	19.5	19.1	18.7	17.9
PPL	19.2	18.8	18.5	17.9
BLEU	19.3	18.8	18.5	17.9
LPU	19.2	18.8	18.5	17.8
Proposed	**18.3**	**18.2**	**18.2**	**17.3**

Table 3: Word Error Rate (WER) with web adapted models for different number of initial sentences.

References

Abhinav Sethy and Panayiotis Georgiou et al.. Building topic specific language models from web-data using competitive models. Proceedings of Eurospeech. 2005

Bing Liu and Yang Dai et al.. Building Text Classifiers Using Positive and Unlabeled Examples. Proceedings of ICDM. 2003

Kamal Nigam and Andrew Kachites McCallum et al.. Text Classification from Labeled and Unlabeled Documents using EM. Journal of Machine Learning. 39(2:3)103–134. 2000

Mirella Lapata and Frank Keller. Web-based models for natural language processing. ACM Transactions on Speech and Language Processing. 2(1),2005.

Philip Resnik and Noah A. Smith. The Web as a parallel corpus. Computational Linguistics. 29(3),2003.

P.G. Georgiou and S.Narayanan et al.. Transonics: A speech to speech system for English-Persian Interactions. Proceedings of IEEE ASRU. 2003

Ruhi Sarikaya and Agustin Gravano et al. Rapid Language Model Development Using External Resources For New Spoken Dialog Domains Proceedings of ICASSP. 2005

Tim Ng and Mari Ostendorf et al.. Web-data Augmented Language Model for Mandarin Speech Recognition. Proceedings of ICASSP. 2005

Xiaojin Zhu. Semi-Supervised Learning Literature Survey. Computer Science, University of Wisconsin-Madison.

A Comparison of Tagging Strategies for Statistical Information Extraction

Christian Siefkes
Database and Information Systems Group, Freie Universität Berlin
Berlin-Brandenburg Graduate School in Distributed Information Systems
Takustr. 9, 14195 Berlin, Germany
siefkes@mi.fu-berlin.de

Abstract

There are several approaches that model *information extraction* as a token classification task, using various *tagging strategies* to combine multiple tokens. We describe the tagging strategies that can be found in the literature and evaluate their relative performances. We also introduce a new strategy, called *Begin/After tagging* or *BIA*, and show that it is competitive to the best other strategies.

1 Introduction

The purpose of *information extraction* (IE) is to find desired pieces of information in natural language texts and store them in a form that is suitable for automatic querying and processing. IE requires a predefined output representation (*target structure*) and only searches for facts that fit this representation. Simple target structures define just a number of *slots* to be filled with a string extracted from a text (*slot filler*). For this simple kind of information extraction, statistical approaches that model IE as a *token classification* task have proved very successful. These systems split a text into a series of tokens and invoke a trainable classifier to decide for each token whether or not it is part of a slot filler of a certain type. To re-assemble the classified tokens into multi-token slot fillers, various *tagging strategies* can be used.

So far, each classification-based IE approach combines a specific tagging strategy with a specific classification algorithm and specific other parameter settings, making it hard to detect how each of these choices influences the results. To allow systematic research into these choices, we have designed a generalized IE system that allows utilizing any tagging strategy with any classification algorithm. This makes it possible to compare strategies or algorithms in an identical setting. In this paper, we describe the tagging strategies that can be found in the literature and evaluate them in the context of our framework. We also introduce a new strategy, called *Begin/After tagging* or *BIA*, and show that it is competitive to the best other strategies. While there are various approaches that employ a classification algorithm with one of the tagging strategies described below, there are no other comparative analyses of tagging strategies yet, to the best of our knowledge.

In the next section, we describe how IE can be modeled as a token classification task and explain the tagging strategies that can be used for this purpose. In Sec. 3 we describe the IE framework and the experimental setup used for comparing the various tagging strategies. In Sec. 4 we list and analyze the results of the comparison.

2 Modeling Information Extraction as a Token Classification Task

There are multiple approaches that model IE as a token classification task, employing standard

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 149–152,
New York, June 2006. ©2006 Association for Computational Linguistics

Strategy	**Triv**	**IOB2**	**IOB1**	**BIE**	**BIA**	**BE**
Special class for first token	–	+	(+)[a]	+	+	+
Special class for last token	–	–	–	+	–	+
Special class for token after last	–	–	–	–	+	–
Number of classes	$n+1$	$2n+1$	$2n+1$	$4n+1$	$3n+1$	$2 \times (n+1)$
Number of classifiers	1	1	1	1	1	2

[a] Only if required for disambiguation

Table 1: Properties of Tagging Strategies

classification algorithms. These systems split a text into a series of tokens and invoke a trainable classifier to decide for each token whether or not it is part of a slot filler of a certain type. To reassemble the classified tokens into multi-token slot fillers, various *tagging strategies* can be used.

The trivial (*Triv*) strategy would be to use a single class for each slot type and an additional "O" class for all other tokens. However, this causes problems if two entities of the same type immediately follow each other, e.g. if the names of two *speakers* are separated by a linebreak only. In such a case, both names would be collapsed into a single entity, since the trivial strategy lacks a way to mark the begin of the second entity.

For this reason (as well as for improved classification accuracy), various more complex strategies are employed that use distinct classes to mark the first and/or last token of a slot filler. The two variations of *IOB* tagging are probably most common: the variant usually called *IOB2* classifies each token as the begin of a slot filler of a certain type (B-*type*), as a continuation of the previously started slot filler, if any (I-*type*), or as not belonging to any slot filler (O). The *IOB1* strategy differs from *IOB2* in using B-*type* only if necessary to avoid ambiguity (i.e. if two same-type entities immediately follow each other); otherwise I-*type* is used even at the beginning of slot fillers. While the *Triv* strategy uses only $n+1$ classes for n slot types, *IOB* tagging requires $2n+1$ classes.

BIE tagging differs from *IOB* in using an additional class for the last token of each slot filler. One class is used for the first token of a slot filler (B-*type*), one for inner tokens (I-*type*) and another one for the last token (E-*type*). A fourth class BE-*type* is used to mark slot fillers consisting of a single token (which is thus both begin and end). *BIE* requires $4n+1$ classes.

A disadvantage of the *BIE* strategy is the high number of classes it uses (twice as many as *IOB1|2*). This can be addressed by introducing a new strategy, *BIA* (or *Begin/After* tagging). Instead of using a separate class for the last token of a slot filler, *BIA* marks the first token *after* a slot filler as A-*type* (unless it is the begin of a new slot filler). Begin (B-*type*) and continuation (I-*type*) of slot fillers are marked in the same way as by *IOB2*. *BIA* requires $3n+1$ classes, n less than *BIE* since no special treatment of single-token slot fillers is necessary.

The strategies discussed so far require only a single classification decision for each token. Another option is to use two separate classifiers, one for finding the begin and another one for finding the end of slot fillers. *Begin/End* (*BE*) tagging requires $n+1$ classes for each of the two classifiers (B-*type* + O for the first, E-*type* + O for the second). In this case, there is no distinction between inner and outer (other) tokens. Complete slot fillers are found by combining the most suitable begin/end pairs of the same type, e.g. by taking the length distribution of slots into account. Table 1 lists the properties of all strategies side by side.

3 Classification Algorithm and Experimental Setup

Our generalized IE system allows employing any classification algorithm with any tagging strategy and any context representation, provided that a suitable implementation or adapter exists. For this paper, we have used the *Winnow* (Littlestone, 1988) classification algorithm and

Strategy	IOB2	IOB1	Triv	BIE	BIA	BE
Seminar Announcements						
etime	97.1	92.4	92.0	94.4	**97.3**	93.6
location	81.7	**81.9**	81.6	77.8	**81.9**	82.3
speaker	85.4	82.0	82.0	84.2	**86.1**	83.7
stime	**99.3**	97.9	97.7	98.6	**99.3**	99.0
Corporate Acquisitions						
acqabr	55.0	53.8	53.9	48.3	**55.2**	50.2
acqloc	27.4	**29.3**	**29.3**	15.7	27.4	18.0
acquired	53.5	**55.7**	55.5	54.8	53.6	53.7
dlramt	71.7	71.5	**71.9**	71.0	71.7	70.5
purchabr	**58.1**	56.1	57.0	47.3	58.0	51.8
purchaser	55.7	55.3	**56.2**	52.7	55.7	55.5
seller	31.8	32.7	**34.7**	27.3	30.1	32.5
sellerabr	25.8	28.0	**28.9**	16.8	24.4	21.4
status	56.9	**57.4**	56.8	56.1	**57.4**	55.2

Table 2: F Percentages for Batch Training

Strategy	IOB1	Triv	BIE	BIA	BE
etime	o (81.6%, –)	o (85.3%, –)	– (98.4%, –)	o (68.6%, +)	o (90.6%, –)
location	o (84.3%, –)	o (90.5%, –)	– (98.9%, –)	o (55.8%, +)	– (98.7%, –)
speaker	– (98.1%, –)	– (95.3%, –)	o (46.7%, –)	o (1.4%, –)	o (20.8%, –)
stime	o (92.9%, –)	– (96.9%, –)	o (75.9%, –)	o (0.0%, =)	o (85.4%, –)
acqabr	o (19.8%, –)	o (12.7%, +)	– (98.8%, –)	o (2.2%, +)	– (99.4%, –)
acqloc	o (75.0%, –)	o (77.8%, –)	– (98.1%, –)	o (11.2%, –)	– (99.3%, –)
acquired	o (17.7%, +)	o (33.6%, +)	o (9.0%, –)	o (0.3%, –)	o (8.9%, +)
dlramt	o (6.6%, –)	o (6.5%, –)	o (5.3%, –)	o (2.9%, –)	o (15.1%, +)
purchabr	o (45.1%, –)	o (37.8%, –)	– (99.9%, –)	o (14.7%, +)	o (94.0%, –)
purchaser	o (62.1%, –)	o (54.8%, –)	o (87.3%, –)	o (6.6%, –)	o (33.8%, –)
seller	o (64.3%, +)	o (72.1%, +)	o (20.1%, –)	o (2.8%, –)	o (24.6%, –)
sellerabr	o (68.0%, +)	o (64.9%, +)	o (91.9%, –)	o (0.8%, –)	o (45.2%, –)
status	o (68.8%, –)	o (70.7%, –)	o (71.7%, –)	o (18.5%, +)	o (64.7%, –)

Table 3: Incremental Training: Significance of Changes Compared to *IOB2*

Strategy	IOB1	Triv	BIE	BIA	BE
etime	o (87.3%, –)	o (91.8%, –)	o (95.0%, –)	o (18.5%, +)	– (96.9%, –)
location	o (18.8%, +)	o (0.5%, –)	– (98.9%, –)	o (22.4%, +)	o (50.3%, +)
speaker	– (98.0%, –)	– (99.1%, –)	o (67.0%, –)	o (55.2%, +)	o (88.8%, –)
stime	o (82.9%, –)	o (84.4%, –)	o (82.2%, –)	o (11.5%, –)	o (73.4%, –)
acqabr	o (49.7%, –)	o (45.8%, –)	– (99.7%, –)	o (6.8%, +)	– (97.9%, –)
acqloc	o (56.3%, +)	o (54.0%, +)	– (99.9%, –)	o (1.1%, +)	– (99.4%, –)
acquired	o (91.5%, +)	o (84.8%, +)	o (67.9%, +)	o (3.5%, +)	o (8.4%, +)
dlramt	o (5.7%, –)	o (14.3%, +)	o (30.2%, –)	o (3.3%, +)	o (46.9%, –)
purchabr	o (77.1%, –)	o (44.0%, –)	– (100.0%, –)	o (6.6%, –)	– (99.5%, –)
purchaser	o (24.1%, –)	o (26.3%, +)	– (96.0%, –)	o (2.5%, –)	o (17.5%, –)
seller	o (34.8%, +)	o (83.5%, +)	– (96.2%, –)	o (59.2%, –)	o (36.1%, +)
sellerabr	o (66.7%, +)	o (76.1%, +)	– (99.7%, –)	o (40.7%, –)	o (90.7%, –)
status	o (26.3%, +)	o (1.5%, –)	o (43.2%, –)	o (28.0%, +)	o (76.0%, –)

Table 4: Batch Training: Significance of Changes Compared to *IOB2*

the context representation described in (Siefkes, 2005), varying only the tagging strategy. An advantage of Winnow is its supporting *incremental* training as well as *batch* training. For many "real-life" applications, automatic extractions will be checked and corrected by a human revisor, as automatically extracted data will always contain errors and gaps that can be detected by human judgment only. This correction process continually provides additional training data, but the usual batch-trainable algorithms are not very suited to integrate new data, since full retraining takes a long time.

We have compared the described tagging strategies on two corpora that are used very often to evaluate IE systems, *CMU Seminar Announcements* and *Corporate Acquisitions*.[1] For both corpora, we used the standard setup: 50/50 training/evaluation split, averaging results over five (Seminar) or ten (Acquisitions) random splits, "one answer per slot" (cf. Lavelli et al. (2004)). Extraction results are evaluated in the usual way by calculating *precision P* and *recall R* of the extracted slot fillers and combining them in the *F-measure*, the harmonic mean of precision and recall: $F = \frac{2 \times P \times R}{P+R}$.[2] For significance testing, we applied a paired two-tailed Student's T-test on the F-measure results, without assuming the variance of the two samples to be equal.

4 Comparison Results

Table 2 list the F-measure results (in percent) reached for both corpora using batch training. Incremental results have been omitted due to lack of space—they are generally slightly worse than batch results, but in many cases the difference is small. For the *Corporate Acquisitions*, the batch results of the best strategies (IOB2 and BIA) are better than any other published results we are aware of; for the *Seminar Announcements*, they are only beaten by the *ELIE* system (Finn and Kushmerick, 2004).[3]

Tables 3 and 4 analyze the performance of each tagging strategy for both training regimes,

[1] Both available from the *RISE Repository* <http://www.isi.edu/info-agents/RISE/>.

[2] This is more appropriate than measuring raw token classification accuracy due to the very unbalanced class distribution among tokens. In the *Seminar Announcements* corpus, our tokenization schema yields 139,021 tokens, only 9820 of which are part of slot fillers. Thus most strategies could already reach an accuracy of 93% by always predicting the `O` class. Also, correctly extracting slot fillers is the goal of IE—a higher token classification accuracy won't be of any use if information extraction performance suffers.

[3] cf. (Siefkes and Siniakov, 2005, Sec. 6.5)

using the popular *IOB2* strategy as a baseline. The first item in each cell indicates whether the strategy performs significantly better ("+") or worse ("–") than *IOB2* or whether the performance difference is not significant at the 95% level ("o"). In brackets, we show the significance of the comparison and whether the results are better or worse when significance is ignored.

Considering these results, we see that the *IOB2* and *BIA* strategies are best. No strategy is able to significantly beat the *IOB2* strategy on any slot, neither with incremental nor batch training. The newly introduced *BIA* strategy is the only one that is able to compete with *IOB2* on all slots. The *IOB1* and *Triv* strategies come close, being significantly worse than *IOB2* only for one or two slots. The two-classifier *BE* strategy is weaker, being significantly outperformed on three (incremental) or four (batch) slots. Worst results are reached by the *BIE* strategy, where the difference is significant in about half of all cases. The good performance of *BIA* is interesting, since this strategy is new and has never been used before (to our knowledge). The *Triv* strategy would have supposed to be weaker, considering how simple this strategy is.

5 Conclusion

Previously, classification-based approaches to IE have combined a specific tagging strategy with a specific classification algorithm and specific other parameter settings, making it hard to detect how each of these choices influences the results. We have designed a generalized IE system that allows exploring each of these choices in isolation. For this paper, we have tested the tagging strategies that can be found in the literature. We have also introduced a new tagging strategy, *BIA* (*Begin/After* tagging).

Our results indicate that the choice of a tagging strategy, while not crucial, should not be neglected when implementing a statistical IE system. The *IOB2* strategy, which is very popular, having been used in public challenges such as those of *CoNLL* (Tjong Kim Sang and De Meulder, 2003) and *JNLPBA* (Kim et al., 2004), has been found to be indeed the best of all established tagging strategies. It is rivaled by the new *BIA* strategy. In typical situations, using one of those strategies should be a good choice—since *BIA* requires more classes, it makes sense to prefer *IOB2* when in doubt.

Considering that it is not much worse, the *Triv* strategy which requires only a single class per slot type might be useful in situations where the number of available classes is limited or the space or time overhead of additional classes is high. The two-classifier *BE* strategy is still interesting if used as part of a more refined approach, as done by the *ELIE* system (Finn and Kushmerick, 2004).[4] Future work will be to observe how well these results generalize in the context of other classifiers and other corpora. To combine the strengths of different tagging strategies, ensemble meta-strategies utilizing the results of multiple strategies could be explored.

References

Aidan Finn and Nicholas Kushmerick. 2004. Multilevel boundary classification for information extraction. In *ECML 2004*, pages 111–122.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *BioNLP/NLPBA 2004*.

A. Lavelli, M. Califf, F. Ciravegna, D. Freitag, C. Giuliano, N. Kushmerick, and L. Romano. 2004. A critical survey of the methodology for IE evaluation. In *LREC*.

Nick Littlestone. 1988. Learning quickly when irrelevant attributes abound. *Machine Learning*, 2.

Christian Siefkes and Peter Siniakov. 2005. An overview and classification of adaptive approaches to information extraction. *Journal on Data Semantics*, IV:172–212. LNCS 3730.

Christian Siefkes. 2005. Incremental information extraction using tree-based context representations. In *CICLing 2005*, LNCS 3406. Springer.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *CoNLL-2003*.

[4] They augment the *BE* strategy with a second level of begin/end classifiers for finding suitable tags matching left-over tags from the level-1 classifiers.

Unsupervised Induction of Modern Standard Arabic Verb Classes

Neal Snider
Linguistics Department
Stanford University
Stanford, CA 94305
snider@stanford.edu

Mona Diab
Center for Computational Learning Systems
Columbia University
New York, NY 10115
mdiab@cs.columbia.edu

Abstract

We exploit the resources in the Arabic Treebank (ATB) for the novel task of automatically creating lexical semantic verb classes for Modern Standard Arabic (MSA). Verbs are clustered into groups that share semantic elements of meaning as they exhibit similar syntactic behavior. The results of the clustering experiments are compared with a gold standard set of classes, which is approximated by using the noisy English translations provided in the ATB to create Levin-like classes for MSA. The quality of the clusters is found to be sensitive to the inclusion of information about lexical heads of the constituents in the syntactic frames, as well as parameters of the clustering algorithm . The best set of parameters yields an $F_{\beta=1}$ score of 0.501, compared to a random baseline with an $F_{\beta=1}$ score of 0.37.

1 Introduction

The creation of the Arabic Treebank (ATB) facilitates corpus based studies of many interesting linguistic phenomena in Modern Standard Arabic (MSA).[1] The ATB comprises manually annotated morphological and syntactic analyses of newswire text from different Arabic sources. We exploit the ATB for the novel task of automatically creating lexical semantic verb classes for MSA. We are interested in the problem of classifying verbs in MSA into groups that share semantic elements of meaning as they exhibit similar syntactic behavior. This manner of classifying verbs in a language is mainly advocated by Levin (1993). The Levin Hypothesis (LH) contends that verbs that exhibit similar syntactic behavior share element(s) of meaning. There exists a relatively extensive classification of English verbs according to different syntactic alternations, and numerous linguistic studies of other languages illustrate that LH holds cross linguistically, in spite of variations in the verb class assignment (Guerssel et al., 1985).

For MSA, the only test of LH has been the work of Mahmoud (1991), arguing for Middle and Unaccusative alternations in Arabic. To date, no general study of MSA verbs and alternations exists. We address this problem by automatically inducing such classes, exploiting explicit syntactic and morphological information in the ATB.

Inducing such classes automatically allows for a large-scale study of different linguistic phenomena within the MSA verb system, as well as cross-linguistic comparison with their English counterparts. Moreover, drawing on generalizations yielded by such a classification could potentially be useful in several NLP problems such as Information Extraction, Event Detection, Information Retrieval and Word Sense Disambiguation, not to mention the facilitation of lexical resource creation such as MSA WordNets and ontologies.

2 Related Work

Based on the Levin classes, many researchers attempt to induce such classes automatically (Merlo and Stevenson, 2001; Schulte im Walde, 2000) . Notably, in the work of Merlo and Stevenson , they attempt to induce three main English verb classes on a large scale from parsed corpora, the class of Unerga-

[1] http://www.ldc.org

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 153–156,
New York, June 2006. ©2006 Association for Computational Linguistics

tive, Unaccusative, and Object-drop verbs. They report results of 69.8% accuracy on a task whose baseline is 34%, and whose expert-based upper bound is 86.5%. In a task similar to ours except for its use of English, Schulte im Walde clusters English verbs semantically by using their alternation behavior, using frames from a statistical parser combined with WordNet classes. She evaluates against the published Levin classes, and reports that 61% of all verbs are clustered into correct classes, with a baseline of 5%.

3 Clustering

We employ both soft and hard clustering techniques to induce the verb classes, using the clustering algorithms implemented in the library *cluster* (Kaufman and Rousseeuw, 1990) in the *R* statistical computing language. The soft clustering algorithm, called FANNY, is a type of fuzzy clustering, where each observation is "spread out" over various clusters. Thus, the output is a membership function $P(x_i, c)$, the membership of element x_i to cluster c. The memberships are nonnegative and sum to 1 for each fixed observation. The algorithm takes k, the number of clusters, as a parameter and uses a Euclidean distance measure.

The hard clustering used is a type of *k*-means clustering The canonical *k*-means algorithm proceeds by iteratively assigning elements to a cluster whose center (centroid) is closest in Euclidian distance.

4 Features

For both clustering techniques, we explore three different sets of features. The features are cast as the column dimensions of a matrix with the MSA lemmatized verbs constituting the row entries.

Information content of frames This is the main feature set used in the clustering algorithm. These are the syntactic frames in which the verbs occur. The syntactic frames are defined as the sister constituents of the verb in a Verb Phrase (VP) constituent.

We vary the type of information resulting from the syntactic frames as input to our clustering algorithms. We investigate the impact of different levels of granularity of frame information on the clustering of the verbs. We create four different data sets based on the syntactic frame information reflecting four levels of frame information: FRAME1 includes all frames with all head information for PPs and SBARs, FRAME2 includes only head information for PPs but no head information for SBARs, FRAME3 includes no head information for neither PPs nor SBARs, and FRAME4 is constructed with all head information, but no constituent ordering information. For all four frame information sets, the elements in the matrix are the co-occurrence frequencies of a verb with a given column heading.

Verb pattern The ATB includes morphological analyses for each verb resulting from the Buckwalter [2] analyzer. Semitic languages such as Arabic have a rich templatic morphology, and this analysis includes the root and pattern information of each verb. This feature is of particular scientific interest because it is unique to the Semitic languages, and has an interesting potential correlation with argument structure.

Subject animacy In an attempt to allow the clustering algorithm to use information closer to actual argument structure than mere syntactic frames, we add a feature that indicates whether a verb requires an animate subject. Following a technique suggested by Merlo and Stevenson , we take advantage of this tendency by adding a feature that is the number of times each verb occurs with each NP types as subject, including when the subject is pronominal or pro-dropped.

5 Evaluation

5.1 Data Preparation

The data used is obtained from the ATB. The ATB is a collection of 1800 stories of newswire text from three different press agencies, comprising a total of $800,000$ Arabic tokens after clitic segmentation. The domain of the corpus covers mostly politics, economics and sports journalism. Each active verb is extracted from the lemmatized treebank along with its sister constituents under the VP. The elements of the matrix are the frequency of the row verb co-occuring with a feature column entry. There are 2074 verb types and 321 frame types, corresponding to 54954 total verb frame tokens. Subject animacy

[2] http://www.ldc.org

information is extracted and represented as four feature columns in our matrix, corresponding to the four subject NP types. The morphological pattern associated with each verb is extracted by looking up the lemma in the output of the morphological analyzer, which is included with the treebank release.

5.2 Gold Standard Data

The gold standard data is created automatically by taking the English translations corresponding to the MSA verb entries provided with the ATB distributions. We use these English translations to locate the lemmatized MSA verbs in the Levin English classes represented in the Levin Verb Index. Thereby creating an approximated MSA set of verb classes corresponding to the English Levin classes. Admittedly, this is a crude manner to create a gold standard set. Given the lack of a pre-existing classification for MSA verbs, and the novelty of the task, we consider it a first approximation step towards the creation of a real gold standard classification set in the near future.

5.3 Evaluation Metric

The evaluation metric used here is a variation on an F-score derived for hard clustering (Rijsbergen, 1979). The result is an F_β measure, where β is the coefficient of the relative strengths of precision and recall. $\beta = 1$ for all results we report. The score measures the maximum overlap between a hypothesized cluster (HYP) and a corresponding gold standard cluster (GOLD), and computes a weighted average across all the HYP clusters: $F_\beta = \sum_{A \in \mathcal{A}} \frac{\|A\|}{V_{tot}} \max_{C \in \mathcal{C}} \frac{(\beta^2 + 1)\|A \cap C\|}{\beta^2\|C\| + \|A\|}$

Here $\mathcal{A}$ is the set of HYP clusters, $\mathcal{C}$ is the set of GOLD clusters, and $V_{tot} = \sum_{A \in \mathcal{A}} \|A\|$ is the total number of verbs that were clustered into the HYP set. This can be larger than the number of verbs to be clustered because verbs can be members of more than one cluster.

5.4 Results

To determine the best clustering of the extracted verbs, we run tests comparing five different parameters of the model, in a $6x2x3x3x3$ design. For the first parameter, we examine six different frame dimensional conditions, FRAME1+ SUBJAnimacy + VerbPatt,FRAME2 + SUBJAnimacy + VerbPatt,FRAME3 + SUBJAnimacy + VerbPatt, FRAME4 + SUBJAnimacy + VerbPatt, FRAME1 + VerbPatt only; and finally, FRAME1+ SUBJAnimacy only . The second parameter is hard vs. soft clustering. The last three conditions are the number of verbs clustered, the number of clusters, and the threshold values used to obtain discrete clusters from the soft clustering probability distribution.

We compare our best results to a random baseline. In the baseline, verbs are randomly assigned to clusters where a random cluster size is on average the same size as each other and as GOLD.[3] The highest overall scored $F_{\beta=1}$ is 0.501 and it results from using FRAME1+SUBJAnimacy+VerbPatt, 125 verbs, 61 clusters, and a threshold of 0.09 in the soft clustering condition. The average cluster size is 3, because this is a soft clustering. The random baseline achieves an overall $F_{\beta=1}$ of 0.37 with comparable settings of 125 verbs randomly assigned to 61 clusters of approximately equal size. A representative mean $F_{\beta=1}$ score is 0.31, and the worst $F_{\beta=1}$ score obtained is 0.188. This indicates that the clustering takes advantage of the structure in the data. To support this observation, a statistical analysis of the clustering experiments is undertaken in the next section.

6 Discussion

For further quantitative error analysis of the data, we perform ANOVAs to test the significance of the differences among the various parameter settings of the clustering algorithm. We find that information type is highly significant ($p < .001$). Within varying levels of the frame information parameter, FRAME2 and FRAME3 are significantly worse than using FRAME1 information ($p < .02$). The effects of SUBJAnimacy, VerbPatt, and FRAME4 are not significantly different from using FRAME1 alone as a baseline, which indicates that these features do not independently contribute to improve clustering, i.e. FRAME1 implicitly encodes the information in VerbPatt and SUBJAnimacy. Also, algorithm type (soft or hard) is found to be significant ($p < .01$),

[3] It is worth noting that this gives an added advantage to the random baseline, since a comparable to GOLD size implicitly contibutes to a higher overlap score.

with soft clustering being better than hard clustering, while controlling for other factors. Among the control factors, verb number is significant ($p < .001$), with 125 verbs being better than both 276 and 407 verbs. The number of clusters is also significant ($p < .001$), with more clusters being better than fewer.

As evident from the results of the statistical analysis, the various informational factors have an interesting effect on the quality of the clusters. Including lexical head information in the frames significantly improves clustering, confirming the intuition that such information is a necessary part of the alternations that define verb classes. However, as long as head information is included, configurational information about the frames does not appear to help the clustering, i.e. ordering of constituents is not significant. It seems that rich Arabic morphology plays a role in rendering order insignificant. Nonetheless, this is an interesting result from a linguistic perspective that begs further investigation. Also interesting is the fact that SUBJAnimacy and the VerbPatt do not help improve clustering. The non-significance of SUBJAnimacy is indeed surprising, given its significant impact on English clusterings. Perhaps the cues utilized in our study require more fine tuning. The lack of significance of the pattern information could indicate that the role played by the patterns is already encoded in the subcategorization frame, therefore pattern information is superfluous.

The score of the best parameter settings with respect to the baseline is considerable given the novelty of the task and lack of good quality resources for evaluation. Moreover, there is no reason to expect that there would be perfect alignment between the Arabic clusters and the corresponding translated Levin clusters, primarily because of the quality of the translation, but also because there is unlikely to be an isomorphism between English and Arabic lexical semantics, as assumed here as a means of approximating the problem.

In an attempt at a qualitative analysis of the resulting clusters, we manually examine several HYP clusters. As an example, one includes the verbs *>aloqaY [meet], $ahid [view], >ajoraY [run an interview], {isotaqobal [receive a guest], Eaqad [hold a conference], >aSodar [issue]*. We note that they all share the concept of convening, or formal meetings. The verbs are clearly related in terms of their event structure (they are all activities, without an associated change of state) yet are not semantically similar. Therefore, our clustering approach yields a classification that is on par with the Levin classes in the coarseness of the cluster membership granularity. In summary, we observe very interesting clusters of verbs which indeed require more in depth lexical semantic study as MSA verbs in their own right.

7 Conclusions

We successfully perform the novel task of applying clustering techniques to verb frame information acquired from the ATB to induce lexical semantic classes for MSA verbs. In doing this, we find that the quality of the clusters is sensitive to the inclusion of information about lexical heads of the constituents in the syntactic frames, as well as parameters of the clustering algorithm. Our classification performs well with respect to a gold standard clusters produced by noisy translations of English verbs in the Levin classes. Our best clustering condition when we use all frame information and the most frequent verbs in the ATB and a high number of clusters outperforms a random baseline by $F_{\beta=1}$ difference of 0.13. This analysis leads us to conclude that the clusters are induced from the structure in the data

Our results are reported with a caveat on the gold standard data. We are in the process of manually cleaning the English translations corresponding to the MSA verbs.

References

M. Guerssel, K. Hale, M. Laughren, B. Levin, and J. White Eagle. 1985. A cross linguistic study of transitivity alternations. In *Papers from the Parasession on Causatives and Agentivity*, volume 21:2, pages 48–63. CLS, Chicago.

L. Kaufman and P.J. Rousseeuw. 1990. *Finding Groups in Data.* John Wiley and Sons, New York.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation.* University of Chicago Press, Chicago.

Abdelgawad T. Mahmoud. 1991. A contrastive study of middle and unaccusative constructions in Arabic and English. In B. Comrie and M. Eid, editors, *Perspectives on Arabic Linguistics*, volume 3, pages 119–134. Benjamins, Amsterdam.

Paola Merlo and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(4).

C.J. Van Rijsbergen. 1979. *Information Retrieval.* Butterworths, London.

Sabine Schulte im Walde. 2000. Clustering verbs semantically according to their alternation behaviour. In *18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrucken, Germany.

Sentence Planning for Realtime Navigational Instructions

Laura Stoia and **Donna K. Byron** and
Darla Magdalene Shockley and **Eric Fosler-Lussier**
The Ohio State University
Computer Science and Engineering
2015 Neil Ave., Columbus, Ohio 43210
stoia|dbyron|shockley|fosler@cse.ohio-state.edu

Abstract

In the current work, we focus on systems that provide incremental directions and monitor the progress of mobile users following those directions. Such directions are based on dynamic quantities like the visibility of reference points and their distance from the user. An intelligent navigation assistant might take advantage of the user's mobility within the setting to achieve communicative goals, for example, by repositioning him to a point from which a description of the target is easier to produce. Calculating spatial variables over a corpus of human-human data developed for this study, we trained a classifier to detect contexts in which a target object can be felicitously described. Our algorithm matched the human subjects with 86% precision.

1 Introduction and Related Work

Dialog agents have been developed for a variety of navigation domains such as in-car driving directions (Dale et al., 2003), tourist information portals (Johnston et al., 2002) and pedestrian navigation (Muller, 2002). In all these applications, the human partner receives navigation instructions from a system. For these domains, contextual features of the physical setting must be taken into account for the agent to communicate successfully.

In dialog systems, one misunderstanding can often lead to additional errors (Moratz and Tenbrink, 2003), so the system must strategically choose instructions and referring expressions that can be clearly understood by the user. Human cognition studies have found that the *in front of/behind* axis is easier to perceive than other relations (Bryant et al., 1992). In navigation tasks, this suggests that describing an object when it is *in front of* the follower is preferable to using other spatial relations. Studies on direction-giving language have found that speakers interleave repositioning commands (e.g. "Turn right 90 degrees") designating objects of interest (e.g. "See that chair?") and action commands (e.g. "Keep going")(Tversky and Lee, 1999). The content planner of a spoken dialog system must decide which of these dialog moves to produce at each turn.

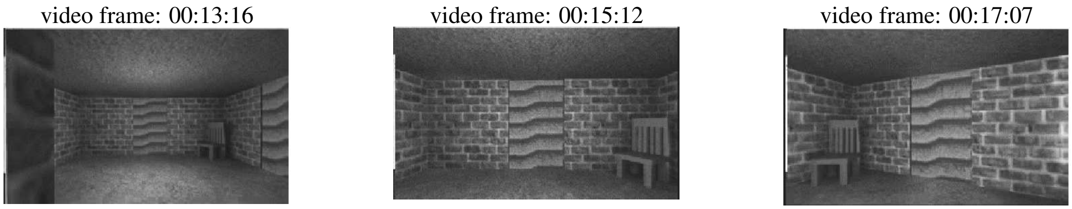
A route plan is a linked list of arcs between nodes representing locations and decision-points in the world. A direction-giving agent must perform several content-planning and surface realization steps, one of which is to decide how much of the route to describe to the user at once (Dale et al., 2003). Thus, the system selects the next target destination and must describe it to the user. In an interactive system, the generation agent must not only decide what to say to the user but also when to say it.

2 Dialog Collection Procedure

Our task setup employs a virtual-reality (VR) world in which one partner, the direction-follower (DF), moves about in the world to perform a series of tasks, such as pushing buttons to re-arrange objects in the room, picking up items, etc. The partners communicated through headset microphones. The simulated world was presented from first-person perspective on a desk-top computer monitor. The DF has no knowledge of the world map or tasks.

His partner, the direction-giver (DG), has a paper 2D map of the world and a list of tasks to complete. During the task, the DG has instant feedback about

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 157–160,
New York, June 2006.



video frame: 00:13:16

00:13:16 “keep going forward”

video frame: 00:15:12

00:14:05 “ok, stop”
00:15:20 “turn right”

video frame: 00:17:07

00:17:19: “and go through *that door* [D6]”

Figure 1: An example sequence with repositioning

DG: ok, yeah, go through *that door* [D9, locate]
turn to your right
’mkay, and there’s *a door* [D11, vague]
in there um, go through *the one straight in front of you* [D11, locate]
ok, stop... and then **turn around and look at *the buttons*** [B18,B20,B21]
ok, you wanna push *the button that’s there on the left by the door* [B18]
ok, and then go through *the door* [D10]
look to your left
there, in *that cabinet there* [C6, locate]

Figure 2: Sample dialog fragment

the DF’s location in the VR world, via mirroring of his partner’s screen on his own computer monitor. The DF can change his position or orientation within the virtual world independently of the DG’s directions, but since the DG knows the task, their collaboration is necessary. In this study, we are most interested in the behavior of the DG, since the algorithm we develop emulates this role. Our paid participants were recruited in pairs, and were self-identified native speakers of North American English.

The video output of DF’s computer was captured to a camera, along with the audio stream from both microphones. A logfile created by the VR engine recorded the DF’s coordinates, gaze angle, and the position of objects in the world. All 3 data sources were synchronized using calibration markers. A technical report is available (Byron, 2005) that describes the recording equipment and software used.

Figure 2 is a dialog fragment in which the DG steers his partner to a cabinet, using both a sequence of target objects and three additional repositioning commands (in bold) to adjust his partner’s spatial relationship with the target.

2.1 Developing the Training Corpus

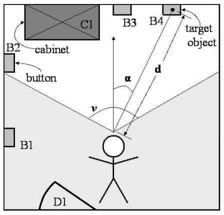
We recorded fifteen dialogs containing a total of 221 minutes of speech. The corpus was transcribed and word-aligned. The dialogs were further annotated using the Anvil tool (Kipp, 2004) to create a set of target referring expressions. Because we are interested in the spatial properties of the referents of these target referring expressions, the items included in this experiment were restricted to objects with a defined spatial position (buttons, doors and cabinets). We excluded plural referring expressions, since their spatial properties are more complex, and also expressions annotated as *vague* or *abandoned*. Overall, the corpus contains 1736 markable items, of which 87 were annotated as vague, 84 abandoned and 228 sets.

We annotated each referring expression with a boolean feature called **Locate** that indicates whether the expression is the first one that allowed the follower to identify the object in the world, in other words, the point at which joint spatial reference was achieved. The kappa (Carletta, 1996) obtained on this feature was 0.93. There were 466 referring expressions in the 15-dialog corpus that were annotated TRUE for this feature.

The dataset used in the experiments is a consensus version on which both annotators agreed on the set of markables. Due to the constraints introduced by the task, referent annotation achieved almost perfect agreement. Annotators were allowed to look ahead in the dialog to assign the referent. The data used in the current study is only the DG’s language.

3 Algorithm Development

The generation module receives as input a route plan produced by a planning module, composed of a list of graph nodes that represent the route. As each subsequent target on the list is selected, content planning considers the tuple of variables <ID, LOC> where ID is an identifier for the target and LOC is the DF’s location (his Cartesian coordinates and orientation angle). Target ID’s are always object id’s to be visited in performing the task, such as a door



v = Visible area(100^o)
α = Angle to target
d = distance to target
In this scene:
Distractors = 5
{B1, B2, B3, C1, D1}
VisDistracts = 3 {B2, B3, C1}
VisSemDistracts = 2 {B2, B3}

Figure 3: An example configuration with spatial context features. The target obje ct is B4 and [B1, B2, B3, B4, C1, D1] are perceptually accessible.

that the DF must pass through. The VR world updates the value of LOC at a rate of 10 frames/sec. Using these variables, the content planner must decide whether the DF's current location is appropriate for producing a referring expression to describe the object.

The following features are calculated from this information: absolute **Angle** between target and follower's view direction, which implicitly gives the *in front* relation, **Distance** from target, visible distractors (**VisDistracts**), visible distractors of the same semantic category (**VisSemDistracts**), whether the target is visible (boolean **Visible**), and the target's semantic category (**Cat**: button/door/cabinet). Figure 3 is an example spatial configuration with these features identified.

3.1 Decision Tree Training

Training examples from the annotation data are tuples containing the ID of the annotated description, the LOC of the DF at that moment (from the VR engine log), and a class label: either Positive or Negative. Because we expect some latency between when the DG judges that a felicity condition is met and when he begins to speak, rather than using spatial context features that co-occur with the onset of each description, we averaged the values over a 0.3 second window centered at the onset of the expression.

Negative contexts are difficult to identify since they often do not manifest linguistically: the DG may say nothing and allow the user to continue moving along his current vector, or he may issue a movement command. A minimal criterion for producing an expression that can achieve joint spatial reference is that the addressee must have perceptual accessibility to the item. Therefore, negative training examples for this experiment were selected from the time-periods that elapsed between the follower achieving perceptual access to the object (coming into the same room with it but not necessarily looking at it), but before the Locating description was spoken. In these negative examples, we consider the basic felicity conditions for producing a descriptive reference to the object to be met, yet the DG did not produce a description. The dataset of 932 training examples was balanced to contain 50% positive and 50% negative examples.

3.2 Decision Tree Performance

This evaluation is based on our algorithm's ability to reproduce the linguistic behavior of our human subjects, which may not be ideal behavior.

The Weka[1] toolkit was used to build a decision tree classifier (Witten and Frank, 2005). Figure 4 shows the resulting tree. 20% of the examples were held out as test items, and 80% were used for training with 10 fold cross validation. Based on training results, the tree was pruned to a minimum of 30 instances per leaf. The final tree correctly classified 86% of the test data.

The number of positive and negative examples was balanced, so the first baseline is 50%. To incorporate a more elaborate baseline, we consider that a description will be made only if the referent is visible to the DF. Marking all cases where the referent was visible as *describe-id* and all the other examples as *delay* gives a higher baseline of 70%, still 16% lower than the result of our tree.[2]

Previous findings in spatial cognition consider angle, distance and shape as the key factors establishing spatial relationships (Gapp, 1995), the angle deviation being the most important feature for projective spatial relationship. Our algorithm also selects **Angle** and **Distance** as informative features. **VisDistracts** is selected as the most important feature by the tree, suggesting that having a large number of objects to contrast makes the description harder, which is in sync with human intuition. We note that Visible is not selected, but that might be due to the fact that it reduces to Angle $> 50^o$. In terms of the referring expression generation algorithm described by (Reiter and Dale, 1992), in which the description which eliminates the most distractors is selected, our

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] not all positive examples were visible

results suggest that the human subjects chose to reduce the size of the distractor set before producing a description, presumably in order to reduce the computational load required to calculate the optimal description.

```
VisDistracts <= 3
| Angle <= 33
| | Distance <=154: describe-id (308/27)
| | Distance > 154: delay (60/20)
| Angle > 33
| | Distance <= 90
| | | Angle <=83:describe-id(79/20)
| | | Angle > 83: delay (53/9)
| | Distance >90: delay(158/16)
VisDistracts > 3: delay (114/1)
```

Figure 4: The decision tree obtained.

Class	Precision	Recall	F-measure
describe-id	0.822	0.925	0.871
delay	0.914	0.8	0.853

Table 1: Detailed Performance

The exact values of features shown in our decision tree are specific to our environment. However, the features themselves are domain-independent and are relevant for any spatial direction-giving task, and their relative influence over the final decision may transfer to a new domain. To incorporate our findings in a system, we will monitor the user's context and plan a description only when our tree predicts it.

4 Conclusions and Future Work

We describe an experiment in content planning for spoken dialog agents that provide navigation instructions. Navigation requires the system and the user to achieve joint reference to objects in the environment. To accomplish this goal human direction-givers judge whether their partner is in an appropriate spatial configuration to comprehend a reference spoken to an object in the scene. If not, one strategy for accomplishing the communicative goal is to steer their partner into a position from which the object is easier to describe.

The algorithm we developed in this study, which takes into account spatial context features replicates our human subject's decision to produce a description with 86%, compared to a 70% baseline based on the visibility of the object. Although the spatial details will vary for other spoken dialog domains, the process developed in this study for producing description dialog moves only at the appropriate times should be relevant for spoken dialog agents operating in other navigation domains.

Building dialog agents for situated tasks provides a wealth of opportunity to study the interaction between context and linguistic behavior. In the future, the generation procedure for our interactive agent will be further developed in areas such as spatial descriptions and surface realization. We also plan to investigate whether different object types in the domain require differential processing, as prior work on spatial semantics would suggest.

5 Acknowledgements

We would like to thank the OSU CSE department for funding this work, our participants in the study and to M. White and our reviewers for useful comments on the paper. We also thank Brad Mellen for building the virtual world.

References

D. J. Bryant, B. Tversky, and N. Franklin. 1992. Internal and external spatial frameworks representing described scenes. *Journal of Memory and Language*, 31:74–98.

D. K. Byron. 2005. The OSU Quake 2004 corpus of two-party situated problem-solving dialogs. Technical Report OSU-CISRC-805-TR57, The Ohio State University Computer Science and Engineering Department, Sept., 2005.

J. Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

R. Dale, S. Geldof, and J. Prost. 2003. CORAL: Using natural language generation for navigational assistance. In M. Oudshoorn, editor, *Proceedings of the 26th Australasian Computer Science Conference*, Adelaide, Australia.

K. Gapp. 1995. Angle, distance, shape, and their relationship to projective relations. Technical Report 115, Universitat des Saarlandes.

M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. 2002. MATCH: An architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pages 376–383.

M. Kipp. 2004. *Gesture Generation by Imitation - From Human Behavior to Computer Character Animation*. Dissertation.com.

R. Moratz and T. Tenbrink. 2003. Instruction modes for joint spatial reference between naive users and a mobile robot. In *Proc. RISSP 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, Special Session on New Methods in Human Robot Interaction.*

C. Muller. 2002. Multimodal dialog in a pedestrian navigation system. In *Proceedings of ISCA Tutorial and Research Workshop on Multi-Modal Dialogue in Mobile Environments.*

E. Reiter and R. Dale. 1992. A fast algorithm for the generation of referring expressions. *COLING.*

B. Tversky and P. U. Lee. 1999. Pictorial and verbal tools for conveying routes. Stade, Germany.

I. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques, 2nd Edition.* Morgan Kaufmann, San Francisco.

Using the Web to Disambiguate Acronyms

Eiichiro Sumita[1, 2]
[1] NiCT
[2] ATR SLC
Kyoto 619-0288, JAPAN
eiichiro.sumita@atr.jp

Fumiaki Sugaya[3]
[3] KDDI R&D Labs

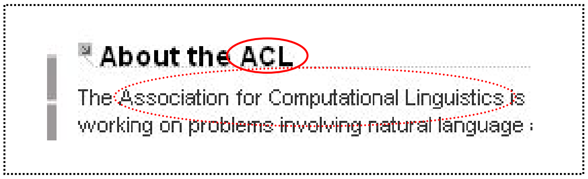
Saitama 356-8502, JAPAN
fsugaya@kddilabs.jp

Abstract

This paper proposes an automatic method for disambiguating an acronym with multiple definitions, considering the context surrounding the acronym. First, the method obtains the Web pages that include both the *acronym* and its *definitions*. Second, the method feeds them to the machine learner. Cross-validation tests results indicate that the current accuracy of obtaining the appropriate definition for an acronym is around 92% for two ambiguous definitions and around 86% for five ambiguous definitions.

1 Introduction

Acronyms are short forms of multiword expressions (we call them *definitions*) that are very convenient and commonly used, and are constantly invented independently everywhere. What each one stands for, however, is often ambiguous. For example, “ACL” has many different definitions, including “Anterior Cruciate Ligament (an injury),” “Access Control List (a concept in computer security),” and “Association for Computational Linguistics (an academic society).” People tend to write acronyms without their definition added nearby (**Table 1**), because acronyms are used to avoid the need to type long expressions. Consequently, there is a strong need to disambiguate acronyms in order to correctly analyze or retrieve text. It is crucial to recognize the correct acronym definition in information retrieval such as a blog search. Moreover, we need to know the meaning of an acronym to translate it correctly. To the best of our knowledge, no other studies have approached this problem.



About the ACL

The Association for Computational Linguistics is working on problems involving natural language

Figure 1 Acronyms and their definitions co-occur in some pages of the Web

On the other side of the coin, an acronym should be defined in its neighborhood. For instance, one may find pages that include a certain *acronym* and its *definition* (**Figure 1**).

First, our proposed method obtains Web pages that include both an *acronym* and its *definitions*. Second, the method feeds them to the machine learner, and the classification program can determine the correct definition according to the context information around the acronym in question.

Definition 1	**Anterior Cruciate Ligament**	**http://www.ehealthmd.com/library/acltears**
She ended up with a torn **ACL**, MCL and did some other damage to her knee. (http://aphotofreak.blogspot.com/2006/01/ill-give-you-everything-i-have-good.html)		
Definition 2	**Access Control List**	**http://en.wikipedia.org/wiki**
Calculating a user’s effective permissions requires more than simply looking up that user’s name in the **ACL**. (http://www.mcsa-exam.com/2006/02/02/effective-permissions.html)		
Definition 3	**Association for Computational Linguistics**	**http://www.aclweb.org/**
It will be published in the upcoming leading **ACL** conference. (http://pahendra.blogspot.com/2005/06/june-14th.html)		

Table 1 Acronym “ACL” without its definition in three different meanings found in blogs

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 161–164,
New York, June 2006. ©2006 Association for Computational Linguistics

Here, we assume that the list of possible definitions for an acronym is given from sources external to this work. Listing pairs of acronyms and their original definitions, on which many studies have been done, such as Nadeau and Turney (2005), results in high performance. Some sites such as http://www.acronymsearch.com/ or http://www.findacronym.com/ provide us with this function.

This paper is arranged as follows. Section 2 explains our solution to the problem, and Section 3 reports experimental results. In Sections 4 and 5 we follow with some discussions and related works, and the paper concludes in Section 6.

2 The proposal

The idea behind this proposal is based on the observation that an *acronym* **often** co-occurs with its *definition* within a single Web page (**Figure 1**). For example, the acronym ACL co-occurs with one of its definitions, "Association for Computational Linguistics," **211,000 times** according to google.com.

Our proposal is a kind of word-sense disambiguation (Pedersen and Mihalcea, 2005). The hit pages can provide us with training data for disambiguating the acronym in question, and the snippets in the pages are fed into the learner of a classifier. Features used in classification will be explained in the latter half of this subsection.

We do not stick to a certain method of machine learning; any state-of-the-art method will suffice. In this paper we employed the decision-tree learning program provided in the WEKA project.

Collecting the training data from the Web

Our input is the acronym in question, A, and the set of its definitions, $\{D_k \mid k=1\sim K\}$.

```
for all k =1~K do
1.Search the Web using query of
   "A AND Dk."
2.Obtain the set of snippets, {Sl
   (A, Dk)| l=1~L}.
3.Separate Dk from Sl and obtain
   the set of training
   data,{(Tl(A), Dk)| l=1~L}.
  End
```

In the experiment, **L is set to 1,000**. Thus, we have for each definition D_k of A, at most 1,000 training data.

Training the classifier

From training data $T_l(A)$, we create feature vectors, which are fed into the learner of the decision tree with correct definition D_k for the acronym A.

Here, we write $T_l(A)$ as $W_{-m}\ W_{-(m-1)}\ ...\ W_{-2}\ W_{-1}\ A\ W_1\ W_2\ ...\ W_{m-1}\ W_m$, where m is from 2 to M, which is called the window size hereafter.

We use keywords within the window of the snippet as features, which are binary, i.e., if the keyword exists in $T_l(A)$, then it is true. Otherwise, it is null.

Keywords are defined in this experiment as the top N frequent words [1], but for A in the bag consisting of all words in $\{T_l(A)\}$. For example, keywords for "ACL" are "*Air, Control, and, Advanced, Agents, MS, Computational, Akumiitti, Cruciate, org, of, CMOS, Language, BOS, Agent, gt, HTML, Meeting, with, html, Linguistics, List, Active, EOS, USA, is, access, Adobe, ACL, ACM, BETA, Manager, list, Proceedings, In, A, League, knee, Anterior, ligament, injuries, reconstruction, injury, on, The, tears, tear, control, as, a, Injury, lt, for, Annual, Association, Access, An, that, this, may, an, you, quot, in, the, one, can, This, by, or, be, to, Logic, 39, are, has, 1, from, middot.*"

3 Experiment

3.1 Acronym and definition preparation

We downloaded a list of acronyms in capital letters only from *Wikipedia* and filtered them by eliminating acronyms shorter than three letters. Then we obtained definitions for each acronym from http://www.acronymsearch.com/ and discarded acronyms that have less than five definitions. Finally, we randomly selected 20 acronyms.

We now have 20 typical acronyms whose ambiguity is more than or equal to five. For each acronym A, a list of definitions $\{ D_k \mid k=1\sim K\ K>=5 \}$, whose elements are ordered by the count of page including A and D_k, is used for the experiment.

[1] In this paper, **N is set to 100**.

3.2 Ambiguity and accuracy

Here we examine the relationship between the degree of ambiguity and classification accuracy by using a cross-validation test for the training data.

#Class	M=2	M=5	M=10	Base
2	**88.7%**	**90.1%**	**92.4%**	82.3%

Table 2 Ambiguity of two

#Class	M=2	M=5	M=10	Base
5	**78.6%**	**82.6%**	**86.0%**	76.5%

Table 3 Ambiguity of five

Ambiguity of two

The first experiment was performed with the selected twenty acronyms by limiting the top two most frequent definitions. **Table 2** summarizes the ten-fold cross validation. While the accuracy changes acronym by acronym, the average is high about 90% of the time. The M in the table denotes the window size, and the longer the window, the higher the accuracy.

The "base" column displays the average accuracy of the baseline method that always picks the most frequent definition. The proposed method achieves better accuracy than the baseline.

Ambiguity of five

Next, we move on to the ambiguity of five (**Table 3**). As expected, the performance is poorer than the abovementioned case, though it is still high, i.e., the average is about 80%. Other than this, our observations were similar to those for the ambiguity of two.

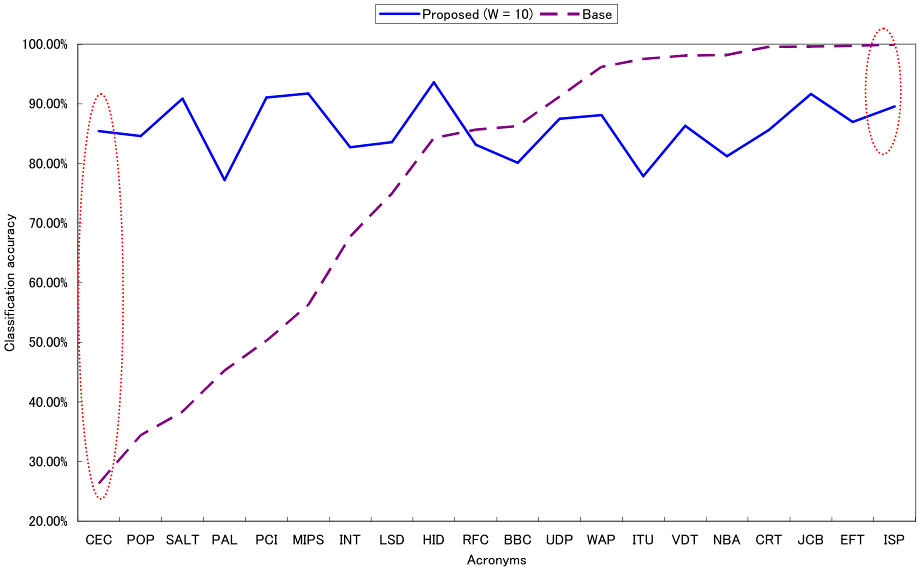


Figure 2 Bias in distribution of definitions (ambiguity of 5)

4 Discussion on biased data

4.1 Problem caused by biased distribution and a countermeasure against it

For some words, the baseline is more accurate than the proposed method because the baseline method reaches all occurrences on the Web thanks to the search engine, whereas our method limits the number of training data by L as mentioned in Section 2. The average quantity of training data was about 830 due to the limit of L, 1,000. The distribution of these training data is rather flat. This causes our classifier to fail in some cases. For example, for the acronym "ISP," the most frequent definition out of five has a share of 99.9% (**Table 4**) on the Web, whereas the distribution in the training data is different from the sharp distribution. Thus, our classification accuracy is not as good as that of the baseline.

Considering the acronym "CEC," the most frequent out of five definitions has the much smaller share of 26.3% on the Web (**Table 5**), whereas the distribution in the training data is similar to the flat distribution of real data. Furthermore, the decision tree learns the classification well, whereas the baseline method performs terribly.

These two extreme cases indicate that for some acronyms, our proposed method is beaten by the baseline method. The slanting line in **Figure 2** shows the baseline performance compared with our proposed method. In the case where our method is strong, the gain is large, and where our method is weak, the reduction is relatively small. The average performance of our proposed method is higher than that of the baseline.

Definition	Page hits
Internet Service Provider	3,590,000
International Standardized Profile	776
Integrated Support Plan	474
Interactive String Processor	287
Integrated System Peripheral control	266

Table 4 Sharp distribution for "ISP"

Definition	Page hits
California Energy Commission	161,000
Council for Exceptional Children	159,000
Commission of the European Communities	138,000
Commission for Environmental Cooperation	77,400
Cation Exchange Capacity	76,400

Table 5 Flat distribution for "CEC"

A possible countermeasure to this problem would be to incorporate prior probability into the learning process.

4.2 Possible dissimilarity of training and real data

The training data used in the above experiment were only the type of snippets that contain **acronyms and their definitions**; there is no guarantee for documents that contain only **acronyms** are similar to the training data. Therefore, learning is not necessarily successful for real data. However, we tested our algorithm for a similar problem introduced in Section 5.1, where we conducted an open test and found a promising result, suggesting that the above-mentioned fear is groundless.

5 Related works

5.1 Reading proper names

The contribution of this paper is to propose a method to use Web pages for a disambiguation task. The method is applicable to different problems such as reading Japanese proper names (Sumita and Sugaya, 2006). Using a Web page containing a name and its syllabary, it is possible to learn how to read proper names with multiple readings in a similar way. The accuracy in our experiment was around 90% for open data.

5.2 The Web as a corpus

Recently, the Web has been used as a corpus in the NLP community, where mainly counts of hit pages have been exploited (Kilgarriff and Grefenstette, 2003). However, our proposal, Web-Based Language Modeling (Sarikaya, 2005), and Bootstrapping Large Sense-Tagged corpora (Mihalcea, 2002) use the content within the hit pages.

6 Conclusion

This paper proposed an automatic method of disambiguating an acronym with multiple definitions, considering the context. First, the method obtains the Web pages that include both the acronym and its definitions. Second, the method feeds them to the learner for classification. Cross-validation test results obtained to date indicate that the accuracy of obtaining the most appropriate definition for an acronym is around 92% for two ambiguous definitions and around 86% for five ambiguous definitions.

References

A. Kilgarriff and G. Grefenstette. 2003. "Introduction to the special issue on the Web as a corpus," Computational Linguistics 29(3): 333-348.

Rada. F. Mihalcea, 2002. "Bootstrapping Large Sense-Tagged Corpora," Proc. of LREC, pp. 1407-1411.

David Nadeau and Peter D. Turney, 2005. "A supervised learning approach to acronym identification," 18th Canadian Conference on Artificial Intelligence, LNAI3501.

Ted Pedersen and Rada. F. Mihalcea, "Advances in Word Sense Disambiguation," tutorial at ACL 2005. http://www.d.umn.edu/~tpederse/WSDTutorial.html.

Ruhi Sarikaya, Hong-kwang Jeff Kuo, and Yuqing Gao, 2005. Impact of Web-Based Language Modeling on Speech Understanding, Proc. of ASRU, pp. 268-271.

Eiichiro Sumita and Fumiaki Sugaya,. 2006. "Word Pronunciation Disambiguation using the Web," Proc. of HLT-NAACL 2006.

Word Pronunciation Disambiguation using the Web

Eiichiro Sumita[1, 2]
[1] NiCT
[2] ATR SLC
Kyoto 619-0288, JAPAN
eiichiro.sumita@atr.jp

Fumiaki Sugaya[3]
[3] KDDI R&D Labs

Saitama 356-8502, JAPAN
fsugaya@kddilabs.jp

Abstract

This paper proposes an automatic method of reading proper names with multiple pronunciations. First, the method obtains Web pages that include both the proper name and its pronunciation. Second, the method feeds them to the learner for classification. The current accuracy is around 90% for open data.

1 Introduction

Within text-to-speech programs, it is very important to deal with heteronyms, that is, words that are spelt the same but that have different readings, e.g. "bow" (a ribbon) and "bow" (of a ship). Reportedly, Japanese text-to-speech programs read sentences incorrectly more than 10 percent of the time. This problem is mainly caused by heteronyms and three studies have attempted to solve it (Yarowsky, 1996; Li and Takeuchi, 1997; and Umemura and Shimizu, 2000).

They assumed that the pronunciation of a word corresponded directly to the sense tag or part-of-speech of that word. In other words, sense tagging and part-of-speech tagging can determine the reading of a word. However, proper names have the same sense tag, for example, “location” for landmarks and the same part-of-speech, the “noun.” Clearly then, reading proper names is outside the scope of previous studies. Also, the proper names of locations, people, organizations, and others are dominant sources of heteronyms. Here, we focus on proper names. Our proposal is similar to previous studies in that both use machine learning. However, previous methods used expensive resources, e.g., a corpus in which words are manually tagged according to their pronunciation. Instead, we propose a method that automatically builds a pronunciation-tagged corpus using the Web as a source of training data for word pronunciation disambiguation.

This paper is arranged as follows. Section 2 proposes solutions, and Sections 3 and 4 report experimental results. We offer our discussion in Section 5 and conclusions in Section 6.

2 The Proposed Methods

It is crucial to correctly read proper names in open-domain text-to-speech programs, for example, applications that read Web pages or newspaper articles. To the best of our knowledge, no other studies have approached this problem. In this paper, we focus on the Japanese language. In this section, we first explain the Japanese writing system (Sections 2.1), followed by our proposal, the basic method (Section 2.2), and the improved method (Section 2.3).

2.1 The Japanese writing system

First, we should briefly explain the modern Japanese writing system. The Japanese language is represented by three scripts:

[i] Kanji, which are characters of Chinese origin;
[ii] Hiragana, a syllabary (reading); and
[iii] Katakana, also a syllabary (reading).

Script	Sample
KANJI	大平
HIRAGANA (reading)	おおだいら
KATAKANA (reading)	オオダイラ

Table 1 Three writings of a single word

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 165–168,
New York, June 2006.

As exemplified in **Table 1**, there are three writings for the word “大平.” The lower two samples are representations of the same pronunciation of “oo daira.”

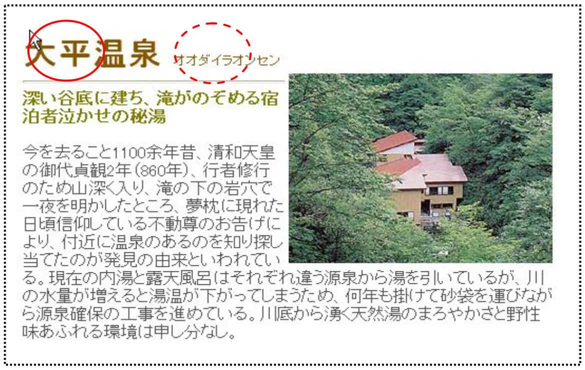
Listing possible readings can be done by consulting a dictionary (see Section 3.1 for the experiment). Therefore, in this paper, we assume that listing is performed prior to disambiguation.

2.2 The basic method based on page hits

The idea is based on the observation that proper names in Kanji often co-occur with their pronunciation in Hiragana (or Katakana) within a single Web page, as shown **Figure 1**. In the figure, the name “大平” in Kanji is indicated with an oval, and its pronunciation in Katakana, “オオダイラ,” is high-lighted with the dotted oval.

According to Google, there are 464 pages in which “大平” and “オオダイラ” co-occur.

In this sense, the co-occurrence frequency suggests to us the most common pronunciation.



大平温泉 オオダイラオンセン

深い谷底に建ち、滝がのぞめる宿泊者泣かせの秘湯

今を去ること1100余年昔、清和天皇の御代貞観2年（860年）、行者修行のため山深く入り、滝の下の岩穴で一夜を明かしたところ、夢枕に現れた日頃信仰している不動尊のお告げにより、付近に温泉のあるのを知り探し当てたのが発見の由来といわれている。現在の内湯と露天風呂はそれぞれ違う源泉から湯を引いているが、川の水量が増えると湯温が下がってしまうため、何年も掛けて砂袋を運びながら源泉確保の工事を進めている。川底から湧く天然湯のまろやかさと野性味あふれる環境は申し分なし。

Figure 1 On the Web, words written in Kanji often co-occur with the pronunciation written in Katakana [1]

Our simple proposal to pick up the most frequent pronunciation achieves surprisingly high accuracy for open data, as Section 4 will later show.

2.3 The improved method using a classifier

The basic method mentioned above merely selects the most frequent pronunciation and neglects all others. This is not disambiguation at all.

The improved method is similar to standard word-sense disambiguation. The hit pages can provide us with training data for reading a particular word. We feed the downloaded data into the learner of a classifier. We do not stick to a certain method of machine learning; any state-of-the-art method will work. The features used in classification will be explained in the latter half of this subsection.

Collecting training data from the Web

Our input is a particular word, W, and the set of its readings, $\{R_k \mid k=1\sim K\}$.

```
For all k =1~K:
  i)   search the Web using the query “W AND $R_k$.”
  ii)  obtain the set of snippets, $\{S_l(W, R_k)| l=1\sim L\}$.
  iii) separate $R_k$ from $S_l$ and obtain the set of training data,$\{(T_l(W), R_k)| l=1\sim L\}$.
  end
```

In the experiments for this report, L is set to 1,000. Thus, for each reading R_k of W, we have, at most 1,000 training data $T_l(W)$.

Training the classifier

From the training data $T_l(W)$, we make feature vectors that are fed into the learner of the decision tree with the correct reading R_k for the word in question, W.

Here, we write $T_l(W)$ as W_{-m} $W_{-(m-1)}$ ... W_{-2} W_{-1} W W_1 W_2 ... W_{m-1} $W_{m,}$ where m is from 2 to M, which hereafter is called the window size.

We use two kinds of features:

- The part-of-speech of W_{-2} W_{-1} and W_1 W_2
- Keywords within the snippet. In this experiment, keywords are defined as the top N frequent words, but for W in the bag consisting of all words in $\{T_l(W)\}$.

In this paper, N is set to 100. These features ground the pronunciation disambiguation task to the real world through the Web. In other words, they give us knowledge about the problem at hand, i.e., how to read proper names in a real-world context.

3 Experimental Data

We conducted the experiments using proper location names.

[1] http://oyudokoro.mimo.com/area/C/cd/tng/000370/index.html

3.1 Ambiguous name lists

Japan Post openly provides postal address lists associated with pronunciations .

From that list, we extracted 79,861 pairs of proper location names and their pronunciations. As the breakdown of **Table 2** shows, 5.7% of proper location names have multiple pronunciations, while 94.3% have a single pronunciation. The average ambiguity is 2.26 for ambiguous types. Next, we took into consideration the frequency of each proper name on the Web. Frequency is surrogated by the page count when the query of a word itself is searched for using a search engine. About one quarter of the occurrences were found to be ambiguous.

Number of readings	type	%
1	70,232	94.3
2	3,443	**5.7**
3	599	
4	150	
5	45	
6	11	
7	4	
8	2	
11	1	
total	74,487	100.0

Table 2 Pronunciation ambiguities in Japanese location names

Our proposal depends on co-occurrences on a Web page. If the pairing of a word W and its reading R do not occur on the Web, the proposal will not work. We checked this, and found that there was only one pair missing out of the 79,861 on our list. In this sense, the coverage is almost 100%.

3.2 Open Data

We tested the performance of our proposed methods on openly available data.

Open data were obtained from the EDR corpus, which consists of sentences from Japanese newspapers. Every word is tagged with part-of-speech and pronunciation.

We extracted sentences that include location heteronyms, that is, those that contain Kanji that can be found in the above-mentioned list of location heteronyms within the postal address data. There were 268 occurrences in total. There were 72 types of heteronyms.

4 Experiment Results

We conducted two experiments: (1) an open test; and (2) a study on the degree of ambiguity.

4.1 Open test

We evaluated our proposals, i.e., the basic method and the improved method with the open data explained in Section 3.1. Both methods achieved a high rate of accuracy.

Basic method performance

In the basic method, the most common pronunciation on the Web is selected. The frequency is estimated by the page count of the query for the pairing of the word W and its pronunciation, R_i.

There are two variations based on the Hiragana and Katakana pronunciation scripts. The average accuracy for the open data was 89.2% for Hiragana and 86.6% for Katakana (**Table 3**). These results are very high, suggesting a strong bias of pronunciation distribution in the open data.

Scripts	Accuracy
HIRAGANA	89.2
KATAKANA	86.6

Table 3 Open test accuracy for the basic method

Performance of the improved method

Table 4 shows the average results for all 268 occurrences. The accuracy of the basic method (**Table 3**) was lower than that of our improved proposal in all window sizes, and it was outperformed at a window size of ten by about 3.5% for both Hiragana and Katakana.

Script	M=2	M=5	M=10
HIRAGANA	89.9	90.3	**92.9**
KATAKANA	89.2	88.4	**89.9**

Table 4 Open test accuracy for the improved method

4.2 Degree of ambiguity

Here, we examine the relationship between the degree of pronunciation ambiguity and pronunciation accuracy using a cross-validation test for training data[2] for the improved method with Hiragana.

Average case

We conducted the first experiment with twenty words[3] that were selected randomly from the Ambiguous Name List (Section 3.1). The average ambiguity was 2.1, indicating the average performance of the improved proposal.

Class	M=2	M=5	M=10	basic
2.1	**89.2 %**	**90.9 %**	**92.3 %**	67.5%

Table 5 Average cases

Table 5 summarizes the ten-fold cross validation, where M in the table is the training data size (window size). The accuracy changes word by word, though the average was high about 90% of the time.

The "basic" column shows the average accuracy of the basic method, i.e., the percentage for the most frequent pronunciation. The improved method achieves much better accuracy than the "basic" one.

The most ambiguous case

Next, we obtained the results (**Table 6**) for the most ambiguous cases, where the degree of ambiguity ranged from six to eleven[4]. The average ambiguity was 7.1.

Class	M=2	M=5	M=10	basic
7.1	**73.9 %**	**77.3 %**	**79.9 %**	57.5%

Table 6 Most ambiguous cases

As we expected, the performances were poorer than the average cases outlined above, although they were still high, i.e., the average ranged from about 70% to about 80 %. Again, the improved method achieved much better accuracy than the "basic" method.[5]

5 Discussion on Transliteration

Transliteration (Knight and Graehl, 1998) is a mapping from one system of writing into another, automation of which has been actively studied between English and other languages such as Arabic, Chinese, Korean, Thai, and Japanese. If there are multiple translation candidates, by incorporating context in a way similar to our proposal, one will be able to disambiguate them.

6 Conclusion

This paper proposed a new method for reading proper names. In our proposed method, using Web pages containing Kanji and Hiragana (or Katakana) representations of the same proper names, we can learn how to read proper names with multiple readings via a state-of-the-art machine learner. Thus, the proposed process requires no human intervention. The current accuracy was around 90% for open data.

References

K. Knight and J. Graehl. 1998 Machine transliteration. Computational Linguistics, 24(4):599-612.

H. Li and J. Takeuchi. 1997. Using Evidence that is both string and Reliable in Japanese Homograph Disambiguation, SIGNL119-9, IPSJ.

Y. Umemura and T. Shimizu. 2000. Japanese homograph disambiguation for speech synthesizers, Toyota Chuo Kenkyujo R&D Review, 35(1):67-74.

D. Yarowsky. 1996. Homograph Disambiguation in Speech Synthesis. In J. van Santen, R. Sproat, J. Olive and J. Hirschberg (eds.), Progress in Speech Synthesis. Springer-Verlag, pp. 159-175.

[2] There is some question as to whether the training data correctly catch all the pronunciations. The experiments in this subsection are independent of this problem, because our intention is to compare the performance of the average case and the most ambiguous case.

[3] 東浜町, 三角町, 宮丸町, 川戸 ,下坂田, 蓬田, 金沢町, 白木町, 神保町, 助谷, 新御堂, 糸原, 駿河町, 百目木, 垣内田町, 杉山町, 百戸, 宝山町, 出来島, 神楽町.

[4] 小谷, 上原町, 上原, 小原, 西原, 上町, 大平, 葛原, 平田, 馬場町, 新田, 土橋町, 大畑町, 上野町, 八幡町, 柚木町, 長田町, 平原.

[5] For some words, the basic accuracy is higher than the cross validation accuracy because the basic method reaches all occurrences on the Web thanks to the search engine, while our improved method limits the number of training data by L in Section 2.3. For example, the most frequent pronunciation of "上原" has 93.7% on the Web, whereas the distribution in the training data is different from such a sharp distribution due to the limitation of L.

Illuminating Trouble Tickets with Sublanguage Theory

Svetlana Symonenko, Steven Rowe, Elizabeth D. Liddy

Center for Natural Language Processing
School Of Information Studies
Syracuse University
Syracuse, NY 13244
{ssymonen, sarowe, liddy}@syr.edu

Abstract

A study was conducted to explore the potential of Natural Language Processing (NLP)-based knowledge discovery approaches for the task of representing and exploiting the vital information contained in field service (trouble) tickets for a large utility provider. Analysis of a subset of tickets, guided by sublanguage theory, identified linguistic patterns, which were translated into rule-based algorithms for automatic identification of tickets' discourse structure. The subsequent data mining experiments showed promising results, suggesting that sublanguage is an effective framework for the task of discovering the historical and predictive value of trouble ticket data.

1 Introduction

Corporate information systems that manage customer reports of problems with products or services have become common nowadays. Yet, the vast amount of data accumulated by these systems remains underutilized for the purposes of gaining proactive, adaptive insights into companies' business operations.

Unsurprising, then, is an increased interest by organizations in knowledge mining approaches to master this information for quality assurance or Customer Relationship Management (CRM) purposes. Recent commercial developments include pattern-based extraction of important entities and relationships in the automotive domain (Attensity, 2003) and text mining applications in the aviation domain (Provalis, 2005).

This paper describes an exploratory feasibility study conducted for a large utility provider. The company was interested in knowledge discovery approaches applicable to the data aggregated by its Emergency Control System (ECS) in the form of field service tickets. When a "problem" in the company's electric, gas or steam distribution system is reported to the corporate Call Center, a new ticket is created. A typical ticket contains the original report of the problem and steps taken to fix it. An operator also assigns a ticket an Original Trouble Type, which can be changed later, as additional information clarifies the nature of the problem. The last Trouble Type assigned to a ticket becomes its Actual Trouble Type.

Each ticket combines structured and unstructured data. The structured portion comes from several internal corporate information systems. The unstructured portion is entered by the operator who receives information over the phone from a person reporting a problem or a field worker fixing it. This free text constitutes the main material for the analysis, currently limited to *known-item* search using keywords and a few patterns. The company management grew dissatisfied with such an approach as time-consuming and, likely, missing out on emergent threats and opportunities or discovering them too late. Furthermore, this approach lacks the ability to knit facts together *across* trouble tickets, except for grouping them by date or gross attributes, such as Trouble Types. The company management felt the need for a system, which, based on the semantic analysis of ticket texts, would not only identify items of interest at a more granular level, such as events, people, locations, dates, relationships, etc., but would also enable the discovery of *unanticipated* associations and trends.

The feasibility study aimed to determine whether NLP-based approaches could deal with

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 169–172,
New York, June 2006. ©2006 Association for Computational Linguistics

such homely, ungrammatical texts and then to explore various knowledge mining techniques that would meet the client's needs. Initial analysis of a sample of data suggested that the goal could be effectively accomplished by looking at the data from the perspective of sublanguage theory.

The novelty of our work is in combining symbolic NLP and statistical approaches, guided by sublanguage theory, which results in an effective methodology and solution for such data.

This paper describes analyses and experiments conducted and discusses the potential of the sublanguage approach for the task of tapping into the value of trouble ticket data.

2 Related Research

Sublanguage theory posits that texts produced within a certain discourse community exhibit shared, often unconventional, vocabulary and grammar (Grishman and Kittredge, 1986; Harris, 1991). Sublanguage theory has been successfully applied in biomedicine (Friedman et al., 2002; Liddy et al., 1993), software development (Etzkorn et al., 1999), weather forecasting (Somers, 2003), and other domains. Trouble tickets exhibit a special discourse structure, combining system-generated, structured data and free-text sections; a special lexicon, full of acronyms, abbreviations and symbols; and consistent "bending" of grammar rules in favor of speed writing (Johnson, 1992; Marlow, 2004). Our work has also been informed by the research on machine classification techniques (Joachims, 2002; Yilmazel et al., 2005).

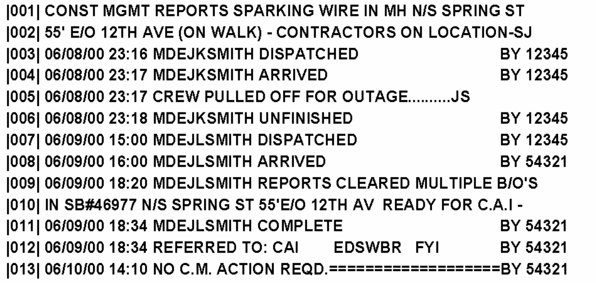
3 Development of the sublanguage model

The client provided us with a dataset of 162,105 trouble tickets dating from 1995 to 2005. An important part of data preprocessing included tokenizing text strings. The tokenizer was adapted to fit the special features of the trouble tickets' vocabulary and grammar: odd punctuation; name variants; domain-specific terms, phrases, and abbreviations.

Development of a sublanguage model began with manual annotation and analysis of a sample of 73 tickets, supplemented with n-gram analysis and contextual mining for particular terms and phrases. The analysis aimed to identify consistent linguistic patterns: domain-specific vocabulary (abbreviations, special terms); major ticket sections; and semantic components (people, organizations, locations, events, important concepts).

The analysis resulted in compiling the core domain lexicon, which includes acronyms for Trouble Types (*SMH* - smoking manhole); departments (*EDS* - Electric Distribution); locations (*S/S/C* - South of the South Curb); special terms (*PACM* - Possible Asbestos Containing Material); abbreviations (*BSMNT* - basement, *F/UP* - follow up); and fixed phrases (*NO LIGHTS, WHITE HAT*). Originally, the lexicon was intended to support the development of the sublanguage grammar, but, since no such lexicon existed in the company, it can now enhance the corporate knowledge base.

Review of the data revealed a consistent structure for trouble ticket discourse. A typical ticket (Fig.1) consists of several text blocks ending with an operator's ID (*12345* or *JS*). A ticket usually opens with a *complaint* (lines *001-002*) that provides the original account of a problem and often contains: reporting entity (*CONST MGMT*), timestamp, short problem description, location. *Field work* (lines *009-010*) normally includes the name of the assigned employee, new information about the problem, steps needed or taken, complications, etc. Lexical choices are limited and section-specific; for instance, reporting a problem typically opens with REPORTS, CLAIMS, or CALLED.



```
|001| CONST MGMT REPORTS SPARKING WIRE IN MH N/S SPRING ST
|002| 55' E/O 12TH AVE (ON WALK) - CONTRACTORS ON LOCATION-SJ
|003| 06/08/00 23:16 MDEJKSMITH DISPATCHED                BY 12345
|004| 06/08/00 23:17 MDEJKSMITH ARRIVED                   BY 12345
|005| 06/08/00 23:17 CREW PULLED OFF FOR OUTAGE.........JS
|006| 06/08/00 23:18 MDEJKSMITH UNFINISHED                BY 12345
|007| 06/09/00 15:00 MDEJLSMITH DISPATCHED                BY 12345
|008| 06/09/00 16:00 MDEJLSMITH ARRIVED                   BY 54321
|009| 06/09/00 18:20 MDEJLSMITH REPORTS CLEARED MULTIPLE B/O'S
|010| IN SB#46977 N/S SPRING ST 55'E/O 12TH AV  READY FOR C.A.I -
|011| 06/09/00 18:34 MDEJLSMITH COMPLETE                  BY 54321
|012| 06/09/00 18:34 REFERRED TO: CAI      EDSWBR   FYI   BY 54321
|013| 06/10/00 14:10 NO C.M. ACTION REQD.===================BY 54321
```

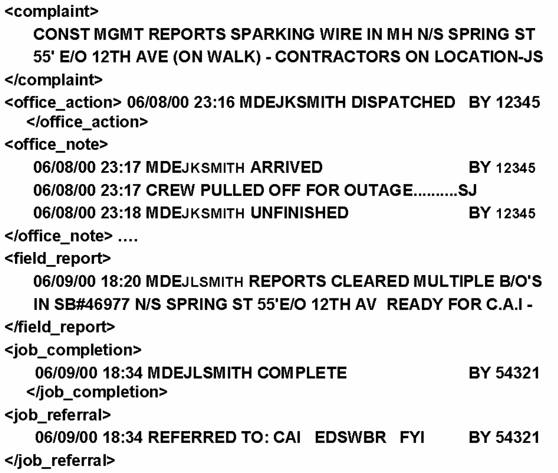
Figure 1. A sample trouble ticket

The resulting typical structure of a trouble ticket (Table 1) includes sections distinct in their content and data format.

Section Name	Data
Complaint	Original report about the problem, Free-text
Office Action	Scheduling actions, Structured text
Office Note	
Field Report	Field work, Free-text
Job Referral	Referring actions, Closing actions, Structured text
Job Completion	
Job Cancelled	

Table 1. Sample discourse structure of a ticket.

Analysis also identified recurring semantic components: people, locations, problem, timestamp, equipment, urgency, etc. The annotation of tickets by sections (Fig.2) and semantic components was validated with domain experts.



```
<complaint>
    CONST MGMT REPORTS SPARKING WIRE IN MH N/S SPRING ST
    55' E/O 12TH AVE (ON WALK) - CONTRACTORS ON LOCATION-JS
</complaint>
<office_action> 06/08/00 23:16 MDEJKSMITH DISPATCHED   BY 12345
   </office_action>
<office_note>
    06/08/00 23:17 MDEJKSMITH ARRIVED                    BY 12345
    06/08/00 23:17 CREW PULLED OFF FOR OUTAGE..........SJ
    06/08/00 23:18 MDEJKSMITH UNFINISHED                 BY 12345
</office_note> ....
<field_report>
    06/09/00 18:20 MDEJLSMITH REPORTS CLEARED MULTIPLE B/O'S
    IN SB#46977 N/S SPRING ST 55'E/O 12TH AV  READY FOR C.A.I -
</field_report>
<job_completion>
    06/09/00 18:34 MDEJLSMITH COMPLETE                   BY 54321
   </job_completion>
<job_referral>
    06/09/00 18:34 REFERRED TO: CAI   EDSWBR   FYI       BY 54321
</job_referral>
```

Figure 2. Annotated ticket sections.

The analysis became the basis for developing logical rules for automatic identification of ticket sections and selected semantic components. Evaluation of system performance on 70 manually annotated and 80 unseen tickets demonstrated high accuracy in automatic section identification, with an error rate of only 1.4%, and no significant difference between results on the annotated vs. unseen tickets. Next, the automatic annotator was run on the entire corpus of 162,105 tickets. The annotated dataset was used in further experiments.

Identification of semantic components brings together variations in names and spellings under a single "normalized" term, thus streamlining and expanding coverage of subsequent data analysis. For example, strings UNSAFE LADDER, HAZ, (hazard) and PACM (Possible Asbestos Containing Material) are tagged and, thus, can be retrieved as *hazard* indicators. "Normalization" is also applied to name variants for streets and departments.

The primary value of the annotation is in effective extraction of structured information from these unstructured free texts. Such information can next be fed into a database and integrated with other data attributes for further analysis. This will significantly expand the range and the coverage of data analysis techniques, currently employed by the company.

The high accuracy in automatic identification of ticket sections and semantic components can, to a significant extent, be explained by the relatively limited number and high consistency of the identified linguistic constructions, which enabled their successful translation into a set of logical rules. This also supported our initial view of the ticket texts as exhibiting sublanguage characteristics, such as: distinct shared common vocabulary and constructions; extensive use of special symbols and abbreviations; and consistent bending of grammar in favor of shorthand. The sublanguage approach thus enables the system to recognize effectively a number of implicit semantic relationships in texts.

4 Leveraging pattern-based approaches with statistical techniques

Next, we assessed the potential of some knowledge discovery approaches to meet company needs and fit the nature of the data.

4.1 Identifying Related Tickets

When several reports relate to the same or recurring trouble, or to multiple problems affecting the same area, a note is made in each ticket, e.g.:

RELATED TO THE 21 ON E38ST TICKET 9999

Each of these related tickets usually contains some aspects of the trouble (Figure 3), but current analytic approaches never brought them together to create a complete picture of the problem, which may provide for useful associations. Semantic component *related-ticket* is expressed through predictable linguistic patterns that can be used as linguistic clues for automatic grouping of related tickets for further analysis.

Ticket 1
..REPORTS FDR-26M49 **OPENED AUTO** @ 16:54..
OTHER TICKETS **RELATED TO THIS JOB**
========= **TICKET 2** =========== **TICKET 3** =

Ticket 2
.. CEILING IS IN VERY BAD CONDITION AND IN
DANGER OFCOLLAPSE. …

Ticket 3
.. CONTRACTOR IS DOING FOUNDATION
WATERPROOFINGWORK ...

Figure 3. Related tickets

4.2 Classification experiments

The analysis of Trouble Type distribution revealed, much to the company's surprise, that 18% of tickets had the Miscellaneous (MSE) Type and, thus, remained out-of-scope for any analysis of associations between Trouble Types and semantic components that would reveal trends. A number of reasons may account for this, including uniqueness of a problem or human error. Review of a sample of MSE tickets showed that some of them should have a more specific Trouble Type. For example (Figure 4), both tickets, each initially assigned the MSE type, describe the WL problem, but only one ticket later receives this code.

Ticket 1 Original Code="**MSE**" Actual Code="**WL**"
WATER LEAKING INTO TRANSFORMER BOX IN BASEMENT OF DORM; …

Ticket 2 Original Code ="**MSE**" Actual Code ="**MSE**"
… WATER IS FLOWING INTO GRADING WHICH LEADS TO ELECTRICIAL VAULT.

Figure 4. *Complaint* sections, WL-problem

Results of n-gram analyses (Liddy et al., 2006), supported our hypothesis that different Trouble Types have distinct linguistic features. Next, we investigated if knowledge of these type-dependent linguistic patterns can help with assigning specific Types to MSE tickets. The task was conceptualized as a multi-label classification, where the system is trained on *complaint* sections of tickets belonging to specific Trouble Types and then tested on tickets belonging either to these Types or to the MSE Type. Experiments were run using the *Extended LibSVM* tool (Chang and Lin, 2001), modified for another project of ours (Yilmazel et al., 2005). Promising results of classification experiments, with precision and recall for known Trouble Types exceeding 95% (Liddy et al., 2006), can, to some extent, be attributed to the fairly stable and distinct language – a sublanguage – of the trouble tickets.

5 Conclusion and Future Work

Initial exploration of the Trouble Tickets revealed their strong sublanguage characteristics, such as: wide use of domain-specific terminology, abbreviations and phrases; odd grammar rules favoring shorthand; and special discourse structure reflective of the communicative purpose of the tickets. The identified linguistic patterns are sufficiently consistent across the data, so that they can be described algorithmically to support effective automated identification of ticket sections and semantic components.

Experimentation with classification algorithms shows that applying the sublanguage theoretical framework to the task of mining trouble ticket data appears to be a promising approach to the problem of reducing human error and, thus, expanding the scope of data amenable to data mining techniques that use Trouble Type information.

Our directions for future research include experimenting with other machine learning techniques, utilizing the newly-gained knowledge of the tickets' sublanguage grammar, as well as testing sublanguage analysis technology on other types of field service reports.

6 References

Improving Product Quality Using Technician Comments.2003. Attensity.

Chang, C.-C. and Lin, C.-J. 2001. *LIBSVM* http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Etzkorn, L. H., Davis, C. G., and Bowen, L. L. 1999. The Language of Comments in Computer Software: A Sublanguage of English. *Journal of Pragmatics, 33*(11): 1731-1756.

Friedman, C., Kraa, P., and Rzhetskya, A. 2002. Two Biomedical Sublanguages: a Description Based on the Theories of Zellig Harris. *Journal of Biomedical Informatics, 35*(4): 222-235.

Grishman, R. and Kittredge, R. I. (Eds.). 1986. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*.

Harris, Z. *A theory of language and information: a mathematical approach.* (1991).

Joachims, T. *Learning to Classify Text using Support Vector Machines: Ph.D. Thesis* (2002).

RFC 1297 - NOC Internal Integrated Trouble Ticket System Functional Specification Wishlist.1992. http://www.faqs.org/rfcs/rfc1297.html.

Liddy, E. D., Jorgensen, C. L., Sibert, E. E., and Yu, E. S. 1993. A Sublanguage Approach to Natural Language Processing for an Expert System. *Information Processing & Management, 29*(5): 633-645.

Liddy, E. D., Symonenko, S., and Rowe, S. 2006. *Sublanguage Analysis Applied to Trouble Tickets.* 19th International FLAIRS Conference.

Marlow, D. 2004. *Investigating Technical Trouble Tickets: An Analysis of a Homely CMC Genre.* HICSS'37.

Application of Statistical Content Analysis Text Mining to Airline Safety Reports.2005. Provalis.

Somers, H. 2003. Sublanguage. In H. Somers (Ed.), *Computers and Translation: A translator's guide.*

Yilmazel, O., Symonenko, S., Balasubramanian, N., and Liddy, E. D. 2005. *Leveraging One-Class SVM and Semantic Analysis to Detect Anomalous Content.* ISI/IEEE'05, Atlanta, GA.

Evolving optimal inspectable strategies for spoken dialogue systems

Dave Toney
School of Informatics
Edinburgh University
2 Buccleuch Place
Edinburgh EH8 9LW
dave@cstr.ed.ac.uk

Johanna Moore
School of Informatics
Edinburgh University
2 Buccleuch Place
Edinburgh EH8 9LW
jmoore@inf.ed.ac.uk

Oliver Lemon
School of Informatics
Edinburgh University
2 Buccleuch Place
Edinburgh EH8 9LW
olemon@inf.ed.ac.uk

Abstract

We report on a novel approach to generating strategies for spoken dialogue systems. We present a series of experiments that illustrate how an *evolutionary* reinforcement learning algorithm can produce strategies that are both optimal and easily inspectable by human developers. Our experimental strategies achieve a mean performance of 98.9% with respect to a predefined evaluation metric. Our approach also produces a dramatic reduction in strategy size when compared with conventional reinforcement learning techniques (87% in one experiment). We conclude that this algorithm can be used to evolve optimal inspectable dialogue strategies.

1 Introduction

Developing a dialogue management strategy for a spoken dialogue system is often a complex and time-consuming task. This is because the number of unique conversations that can occur between a user and the system is almost unlimited. Consequently, a system developer may spend a lot of time anticipating how potential users might interact with the system before deciding on the appropriate system response.

Recent research has focused on generating dialogue strategies automatically. This work is based on modelling dialogue as a markov decision process, formalised by a finite state space *S*, a finite action set *A*, a set of transition probabilities *T* and a reward function *R*. Using this model an optimal dialogue strategy π^* is represented by a mapping between the state space and the action set. That is, for each state $s \in S$ this mapping defines its optimal action a_s^*. How is this mapping constructed? Previous approaches have employed reinforcement learning (RL) algorithms to estimate an optimal value function Q^* (Levin et al., 2000; Frampton and Lemon, 2005). For each state this function predicts the future reward associated with each action available in that state. This function makes it easy to extract the optimal strategy (*policy* in the RL literature).

Progress has been made with this approach but some important challenges remain. For instance, very little success has been achieved with the large state spaces that are typical of real-life systems. Similarly, work on summarising learned strategies for interpretation by human developers has so far only been applied to tasks where each state-action pair is explicitly represented (Lecœuche, 2001). This tabular representation severely limits the size of the state space.

We propose an alternative approach to finding optimal dialogue policies. We make use of XCS, an *evolutionary* reinforcement learning algorithm that seeks to represent a policy as a compact set of state-action rules (Wilson, 1995). We suggest that this algorithm could overcome both the challenge of large state spaces and the desire for strategy inspectability. In this paper, we focus on the issue of inspectability. We present a series of experiments that illustrate how XCS can be used to evolve dialogue strategies that are both optimal and easily inspectable.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 173–176,
New York, June 2006.

2 Learning Classifier Systems and XCS

Learning Classifier Systems were introduced by John Holland in the 1970s as a framework for learning rule-based knowledge representations (Holland, 1976). In this model, a rule base consists of a population of *N* state-action rules known as *classifiers*. The state part of a classifier is represented by a ternary string from the set $\{0,1,\#\}$ while the action part is composed from $\{0,1\}$. The # symbol acts as a wildcard allowing a classifier to aggregate states; for example, the state string 1#1 matches the states 111 and 101. Classifier systems have been applied to a number of learning tasks, including data mining, optimisation and control (Bull, 2004).

Classifier systems combine two machine learning techniques to find the optimal rule set. A genetic algorithm is used to evaluate and modify the population of rules while reinforcement learning is used to assign rewards to existing rules. The search for better rules is guided by the *strength* parameter associated with each classifier. This parameter serves as a fitness score for the genetic algorithm and as a predictor of future reward (*payoff*) for the RL algorithm. This evolutionary learning process searches the space of possible rule sets to find an optimal policy as defined by the reward function.

XCS (X Classifier System) incorporates a number of modifications to Holland's original framework (Wilson, 1995). In this system, a classifier's fitness is based on the accuracy of its payoff prediction instead of the prediction itself. Furthermore, the genetic algorithm operates on actions instead of the population as a whole. These aspects of XCS result in a more complete map of the state-action space than would be the case with strength-based classifier systems. Consequently, XCS often outperforms strength-based systems in sequential decision problems (Kovacs, 2000).

3 Experimental Methodology

In this section we present a simple slot-filling system based on the hotel booking domain. The goal of the system is to acquire the values for three slots: the check-in date, the number of nights the user wishes to stay and the type of room required (single, twin etc.). In slot-filling dialogues, an optimal strategy is one that interacts with the user in a satisfactory way while trying to minimise the length of the dialogue. A fundamental component of user satisfaction is the system's prevention and repair of any miscommunication between it and the user. Consequently, our hotel booking system focuses on evolving essential slot confirmation strategies.

We devised an experimental framework for modelling the hotel system as a sequential decision task and used XCS to evolve three behaviours. Firstly, the system should execute its dialogue acts in a logical sequence. In other words, the system should greet the user, ask for the slot information, present the query results and then finish the dialogue, in that order (Experiment 1). Secondly, the system should try to acquire the slot values as quickly as possible while taking account of the possibility of misrecognition (Experiments 2a and 2b). Thirdly, to increase the likelihood of acquiring the slot values correctly, each one should be confirmed at least once (Experiments 3 and 4).

The reward function for Experiments 1, 2a and 2b was the same. During a dialogue, each non-terminal system action received a reward value of zero. At the end of each dialogue, the final reward comprised three parts: (i) -1000 for each system turn; (ii) 100,000 if all slots were filled; (iii) 100,000 if the first system act was a greeting. In Experiments 3 and 4, an additional reward of 100,000 was assigned if all slots were confirmed.

The transition probabilities were modelled using two versions of a handcoded simulated user. A very large number of test dialogues are usually required for learning optimal dialogue strategies; simulated users are a practical alternative to employing human test users (Scheffler and Young, 2000; Lopez-Cozar et al., 2002). Simulated user A represented a fully cooperative user, always giving the slot information that was asked. User B was less cooperative, giving no response 20% of the time. This allowed us to perform a two-fold cross validation of the evolved strategies.

For each experiment we allowed the system's strategy to evolve over 100,000 dialogues with each simulated user. Dialogues were limited to a maximum of 30 system turns. We then tested each strategy with a further 10,000 dialogues. We logged the total reward (payoff) for each test dialogue. Each experiment was repeated ten times.

In each experiment, the presentation of the query results and closure of the dialogue were combined into a single dialogue act. Therefore, the dialogue acts available to the system for the first experiment were: *Greeting, Query+Goodbye, Ask(Date), Ask(Duration)* and *Ask(RoomType)*. Four boolean variables were used to represent the state of the dialogue: *GreetingFirst*, *DateFilled*, *DurationFilled*, *RoomFilled*.

Experiment 2 added a new dialogue act: *Ask(All)*. The goal here was to ask for all three slot values if the probability of getting the slot values was reasonably high. If the probability was low, the system should ask for the slots one at a time as before. This information was modelled in the simulated users by 2 variables: *Prob1SlotCorrect* and *Prob3SlotsCorrect*. The values for these variables in Experiments 2a and 2b respectively were: 0.9 and 0.729 ($=0.9^3$); 0.5 and 0.125 ($=0.5^3$).

Experiment 3 added three new dialogue acts: *Explicit_Confirm(Date)*, *Explicit_Confirm(Duration)*, *Explicit_Confirm(RoomType)* and three new state variables: *DateConfirmed*, *DurationConfirmed*, *RoomConfirmed*. The goal here was for the system to learn to confirm each of the slot values after the user has first given them. Experiment 4 sought to reduce the dialogue length further by allowing the system to confirm one slot value while asking for another. Two new dialogue acts were available in this last experiment: *Implicit_Confirm(Date)+Ask(Duration)* and *Implicit_Confirm(Duration)+Ask(RoomType)*.

4 Experimental Results

Table 1 lists the total reward (payoff) averaged over the 10 cross-validated test trials for each experiment, expressed as a percentage of the maximum payoff. In these experiments, the maximum payoff represents the shortest possible successful dialogue. For example, the maximum payoff for Experiment 1 is 195,000: 100,000 for filling the slots plus 100,000 for greeting the user at the start of the dialogue minus 5000 for the minimum number of turns (five) taken to complete the dialogue successfully. The average payoff for the 10 trials trained on simulated user A and tested on user B was 193,877 – approximately 99.4% of the maximum possible. In light of these results and the stochastic user responses, we suggest that these evolved strategies would compare favourably with any handcoded strategies.

Exp.	Training/Test Users	Payoff (%)
1	A, B	99.4
	B, A	99.8
2a	A, B	99.1
	B, A	99.4
2b	A, B	96.8
	B, A	97.2
3	A, B	98.8
	B, A	99.3
4	A, B	99.3
	B, A	99.7

Table 1: Payoff results for the evolved strategies.

It is instructive to compare the rate of convergence for different strategies. Figure 1 shows the average payoff for the 100,000 dialogues trained with simulated user A in Experiments 3 and 4. It shows that Experiment 3 approached the optimal policy after approximately 20,000 dialogues whereas Experiment 4 converged after approximately 5000 dialogues. This is encouraging because it suggests that XCS remains focused on finding the shortest successful dialogue even when the number of available actions increases.

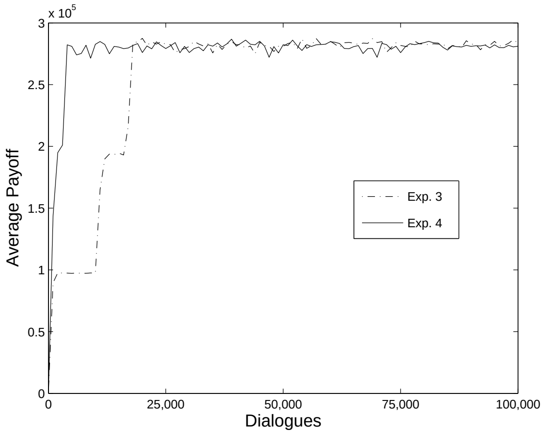


Figure 1: Convergence towards optimality during training in Experiments 3 and 4 (simulated user A).

Finally, we look at how to represent an optimal strategy. From the logs of the test dialogues we extracted the state-action rules (classifiers) that were executed. For example, in Experiment 4, the op-

State							Action
GreetingFirst	*DateFilled*	*DurationFilled*	*RoomFilled*	*DateConfirmed*	*DurationConfirmed*	*RoomConfirmed*	
0	0	#	#	#	#	#	*Greeting*
1	0	0	0	#	#	#	*Ask(Date)*
1	1	#	#	0	#	#	*Implicit_Confirm(Date) + Ask(Duration)*
1	1	1	#	1	0	0	*Implicit_Confirm(Duration) + Ask(RoomType)*
1	1	1	1	1	1	0	*Explicit_Confirm(RoomType)*
1	1	1	1	1	1	1	*Query + Goodbye*

Table 2: A summary of the optimal strategy for Experiment 4.

timal strategy is represented by 17 classifiers. By comparison, a purely RL-based strategy would define an optimal action for every theoretically possible state (i.e. 128). In this example, the evolutionary approach has reduced the number of rules from 128 to 17 (a reduction of 87%) and is therefore much more easily inspectable. In fact, the size of the optimal strategy can be reduced further by selecting the most general classifier for each action (Table 2). These rules are sufficient since they cover the 60 states that could actually occur while following the optimal strategy.

5 Conclusions and future work

We have presented a novel approach to generating spoken dialogue strategies that are both optimal and easily inspectable. The generalizing ability of the evolutionary reinforcement learning (RL) algorithm, XCS, can dramatically reduce the size of the optimal strategy when compared with conventional RL techniques. In future work, we intend to exploit this generalization feature further by developing systems that require much larger state representations. We also plan to investigate other approaches to strategy summarisation. Finally, we will evaluate our approach against purely RL-based methods.

References

Larry Bull, editor. 2004. *Applications of Learning Classifier Systems*. Springer.

Matthew Frampton and Oliver Lemon. 2005. Reinforcement learning of dialogue strategies using the user's last dialogue act. In *IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Edinburgh, UK, July.

John Holland. 1976. Adaptation. In Rosen R. and F. Snell, editors, *Progress in theoretical biology*. Plenum, New York.

Tim Kovacs. 2000. Strength or accuracy? Fitness calculation in learning classifier systems. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart Wilson, editors, *Learning Classifier Systems. From Foundations to Applications*, Lecture Notes in Artificial Intelligence 1813, pages 143–160. Springer-Verlag.

Renaud Lecœuche. 2001. Learning optimal dialogue management rules by using reinforcement learning and inductive logic programming. In *2nd Meeting of the North American Chapter of the Association of Computational Linguistics*, Pittsburgh, USA, June.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialogue strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1):11–23.

R. Lopez-Cozar, A. De la Torre, J. Segura, A. Rubio, and V. Sánchez. 2002. Testing dialogue systems by means of automatic generation of conversations. *Interacting with Computers*, 14(5):521–546.

Konrad Scheffler and Steve Young. 2000. Probabilistic simulation of human-machine dialogues. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1217–1220, Istanbul, Turkey, June.

Stewart Wilson. 1995. Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175.

Lycos Retriever: An Information Fusion Engine

Brian Ulicny

Versatile Information Systems, Inc.
5 Mountainview Drive
Framingham, MA 01701 USA
bulicny@vistology.com

Abstract

This paper describes the Lycos Retriever system, a deployed system for automatically generating coherent topical summaries of popular web query topics.

1 Introduction

Lycos Retriever[1] is something new on the Web: a patent-pending *information fusion engine*. That is, unlike a search engine, rather than returning ranked documents links in response to a query, Lycos Retriever categorizes and disambiguates topics, collects documents on the Web relevant to the disambiguated sense of that topic, extracts paragraphs and images from these documents and arranges these into a coherent summary report or background briefing on the topic at something like the level of the first draft of a Wikipedia[2] article. These topical pages are then arranged into a browsable hierarchy that allows users to find related topics by browsing as well as searching.

2 Motivations

The presentation of search results as ranked lists of document links has become so ingrained that it is hard now to imagine alternatives to it. Other interfaces, such as graphical maps or visualizations, have not been widely adopted. Question-answering interfaces on the Web have not had a high adoption rate, either: it is hard to get users to venture beyond the 2.5 word queries they are accustomed to, and if question-answering results are not reliably better than keyword search, users quickly return to keyword queries. Many user queries specify nothing more than a topic anyway.

But why treat common queries exactly like unique queries? For common queries we know that incentives for ranking highly have led to techniques for artificially inflating a site's ranking at the expense of useful information. So the user has many useless results to sift through. Furthermore, users are responsive to filtered information, as the upsurge in popularity of Wikipedia and Answers.com demonstrate.

Retriever responds to these motivations by automatically generating a narrative summary that answers, "What do I need to know about this topic?" for the most popular topics on the Web.[3]

3 Lycos Retriever pages

Figure 1 shows a sample Retriever page for the topic "Mario Lemieux".[4] The topic is indicated at the upper left. Below it is a category assigned to the topic, in this case *Sports > Hockey > Ice Hockey > National Hockey League > Lemieux, Mario*. The main body of the page is a set of paragraphs beginning with a biographical paragraph complete with Lemieux's birth date, height, weight and position extracted from Nationmaster.com, followed by paragraphs outlining his career from

[1] http://www.lycos.com/retriever.html. Work on Retriever was done while author was employed at Lycos.

[2] http://www.wikipedia.org

[3] See (Liu, 2003) for a similarly motivated system.

[4] For other categories, see e.g. King Kong (1933):
http://www.lycos.com/info/king-kong-1933.html,
Zoloft: http://www.lycos.com/info/zoloft.html,
Public-Key Cryptography: http://www.lycos.com/info/public-key-cryptography.html ,
Lyme Disease: http://www.lycos.com/info/lyme-disease.html,
Reggaeton: http://www.lycos.com/info/reggaeton.html

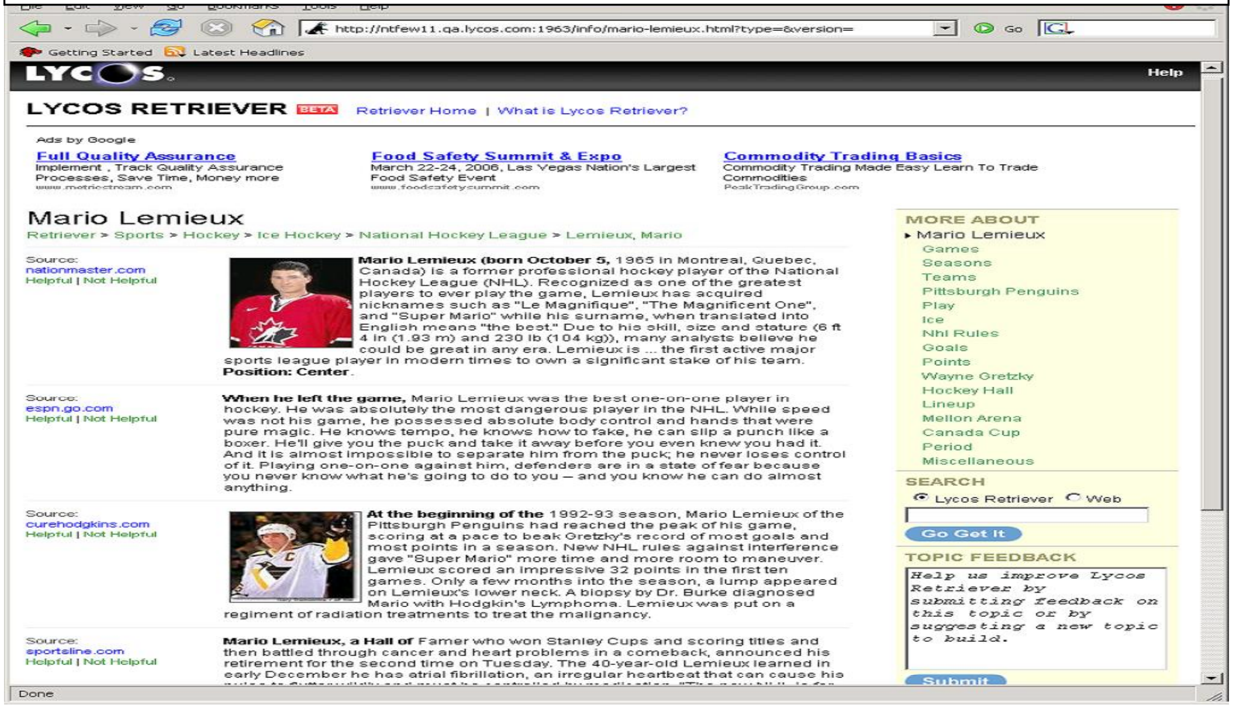
Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 177–180,
New York, June 2006.

other sources. The source for each extract is indicated in shortened form in the left margin of the page; mousing over the shortened URL reveals the full title and URL. Associated images are thumbnailed alongside the extracted paragraphs.

Running down the right side of the page under *More About* is a set of subtopics. Each subtopic is a link to a page (or pages) with paragraphs about the topic (Lemieux) with respect to such subtopics as *Games, Seasons, Pittsburgh Penguins, Wayne Gretzky,* and others, including the unpromising subtopic *ice*.

4 Topic Selection

Figure 1 Retriever Topic Page "Mario Lemieux"


http://ntfew11.qa.lycos.com:1963/info/mario-lemieux.html?type=&version=
Go
Getting Started
Latest Headlines
LYCOS
Help
LYCOS RETRIEVER
BETA
Retriever Home | What is Lycos Retriever?
Go Get It
Submit
Done

Ads by Google

Full Quality Assurance
Implement , Track Quality Assurance Processes, Save Time, Money more
www.metricstream.com

Food Safety Summit & Expo
March 22-24, 2006, Las Vegas Nation's Largest Food Safety Event
www.foodsafetysummit.com

Commodity Trading Basics
Commodity Trading Made Easy Learn To Trade Commodities
PeakTradingGroup.com

Mario Lemieux

Retriever > Sports > Hockey > Ice Hockey > National Hockey League > Lemieux, Mario

Source: nationmaster.com
Helpful | Not Helpful

Mario Lemieux (born October 5, 1965 in Montreal, Quebec, Canada) is a former professional hockey player of the National Hockey League (NHL). Recognized as one of the greatest players to ever play the game, Lemieux has acquired nicknames such as "Le Magnifique", "The Magnificent One", and "Super Mario" while his surname, when translated into English means "the best." Due to his skill, size and stature (6 ft 4 in (1.93 m) and 230 lb (104 kg)), many analysts believe he could be great in any era. Lemieux is ... the first active major sports league player in modern times to own a significant stake of his team. **Position: Center**.

Source: espn.go.com
Helpful | Not Helpful

When he left the game, Mario Lemieux was the best one-on-one player in hockey. He was absolutely the most dangerous player in the NHL. While speed was not his game, he possessed absolute body control and hands that were pure magic. He knows tempo, he knows how to fake, he can slip a punch like a boxer. He'll give you the puck and take it away before you even knew you had it. And it is almost impossible to separate him from the puck; he never loses control of it. Playing one-on-one against him, defenders are in a state of fear because you never know what he's going to do to you — and you know he can do almost anything.

Source: curehodgkins.com
Helpful | Not Helpful

At the beginning of the 1992-93 season, Mario Lemieux of the Pittsburgh Penguins had reached the peak of his game, scoring at a pace to beak Gretzky's record of most goals and most points in a season. New NHL rules against interference gave "Super Mario" more time and more room to maneuver. Lemieux scored an impressive 32 points in the first ten games. Only a few months into the season, a lump appeared on Lemieux's lower neck. A biopsy by Dr. Burke diagnosed Mario with Hodgkin's Lymphoma. Lemieux was put on a regiment of radiation treatments to treat the malignancy.

Source: sportsline.com
Helpful | Not Helpful

Mario Lemieux, a Hall of Famer who won Stanley Cups and scoring titles and then battled through cancer and heart problems in a comeback, announced his retirement for the second time on Tuesday. The 40-year-old Lemieux learned in early December he has atrial fibrillation, an irregular heartbeat that can cause his

MORE ABOUT

- Mario Lemieux
  - Games
  - Seasons
  - Teams
  - Pittsburgh Penguins
  - Play
  - Ice
  - Nhl Rules
  - Goals
  - Points
  - Wayne Gretzky
  - Hockey Hall
  - Lineup
  - Mellon Arena
  - Canada Cup
  - Period
  - Miscellaneous

SEARCH

● Lycos Retriever ○ Web

TOPIC FEEDBACK

Help us improve Lycos Retriever by submitting feedback on this topic or by suggesting a new topic to build.

An initial run of about 60K topics was initiated in December, 2005; this run yielded approximately 30K Retriever topic pages, each of which can have multiple display pages. Retriever topics that had fewer than three paragraphs or which were categorized as pornographic were automatically deleted. The biggest source of topic candidates was Lycos's own query logs. A diverse set of topics was chosen in order to see which types of topics generated the best Retriever pages.

5 Topic Categorization & Disambiguation

After a topic was input to the system, the Retriever system assigned it a category using a naïve Bayes classifier built on a spidered DMOZ[5] hierarchy. Various heuristics were implemented to make the returned set of categories uniform in length and depth, up-to-date, and readable.

Once the categorizer assigned a set of categories to a topic, a disambiguator module determined whether the assigned categories could be assigned to a single thing using a set of disambiguating features learned from the DMOZ data itself. For example, for the topic 'Saturn', the assigned categories included 'Science/Astronomy', 'Recreation/Autos' and 'Computers/Video Games' (Sega Saturn). The disambiguator detected the presence of feature pairs in these that indicated more than one topic. Therefore, it clustered the assigned categories into groups for the car-, astronomy- and video-game-senses of the topic and assigned each group a discriminative term which was used to disambiguate the topic: *Saturn (Auto), Saturn (Solar System), Saturn (Video Game)*. Retriever returned pages only for topics that were believed to be disambiguated according to DMOZ. If no categories

[5] http://www.dmoz.com

were identified via DMOZ, a default *Other* category was assigned unless the system guessed that the topic was a personal name, based on its components.

The live system assigns non-default categories with 86.5% precision; a revised algorithm achieved 93.0% precision, both based on an evaluation of 982 topics. However, our precision on identifying unambiguous topics with DMOZ was only 83%. Still, this compares well with the 75% precision achieved on by the best-performing system on a similar task in the 2005 KDD Cup (Shen 2005).

6 Document Retrieval

After a topic was categorized and disambiguated, the disambiguated topic was used to identify up to 1000 documents from Lycos' search provider. For ambiguous topics various terms were added as optional 'boost' terms, while terms from other senses of the ambiguous topic categories were prohibited. Other query optimization techniques were used to get the most focused document set, with non-English and obscene pages filtered out

7 Passage Extraction

Each URL for the topic was then fetched. An HTML parser converted the document into a sequence of contiguous text blocks. At this point, contiguous text passages were identified as being potentially interesting if they contained an expression of the topic in the first sentence.

When a passage was identified as being potentially interesting, it was then fully parsed to see if an expression denoting the topic was the Discourse Topic of the passage. Discourse Topic is an under-theorized notion in linguistic theory: not all linguists agree that the notion of Discourse Topic is required in discourse analysis at all (cf. Asher, 2004). For our purposes, however, we formulated a set of patterns for identifying Discourse Topics on the basis of the output of the CMU Link Parser[6] the system uses.

Paradigmatically, we counted ordinary subjects of the first sentence of a passage as expressive of the Discourse Topic. So, if we found an expression of the topic there, either in full or reduced form, we took that as an instance of the topic appearing as Discourse Topic in that passage and ranked that passage highly. Of course, not all Discourse Topics are expressed as subjects, and the system recognized this.

A crucial aspect of this functionality is to identify how different sorts of topics can be expressed in a sentence. To give a simple illustration, if the system believes that a topic has been categorized as a personal name, then it accepted reduced forms of the name as expressions of the topic (e.g. "Lindsay" and "Lohan" can both be expressions of the topic "Lindsay Lohan" in certain contexts); but it does not accept reduced forms in all cases.

Paragraphs were verified to contain a sequence of sentences by parsing the rest of the contiguous text. The verb associated with the Discourse Topic of the paragraph was recorded for future use in assembling the topic report. Various filters for length, keyword density, exophoric expressions, spam and obscenity were employed. A score of the intrinsic informativeness of the paragraph was then assigned, making use of such metrics as the length of the paragraph, the number of unique NPs, the type of verb associated with the Discourse Topic, and other factors.

Images were thumbnailed and associated with the extracted paragraph on the basis of matching text in the image filename, alt-text or description elements of the tag as well as the size and proximity of the image to the paragraph at hand. We did not analyze the image itself.

8 Subtopic Selection and Report Assembly

Once the system had an array of extracted paragraphs, ranked by their intrinsic properties, we began constructing the topic report by populating an initial 'overview' portion of the report with some of the best-scoring paragraphs overall.

First, Retriever eliminated duplicate and near-duplicate paragraphs using a spread-activation algorithm.

Next the system applied question-answering methodology to order the remaining paragraphs into a useful overview of the topic: first, we found the best two paragraphs that say *what the topic is*, by finding the best paragraphs where the topic is the Discourse Topic of the paragraph and the associated verb is a copula or copula-like (e.g. *be known as*). Then, in a similar way, we found the best few paragraphs that said *what*

[6] http://www.link.cs.cmu.edu/link/

attributes the topic has. Then, a few paragraphs that said *what the topic does*, followed by a few paragraphs that said *what happens to the topic* (how it is used, things it has undergone, and so on).

The remaining paragraphs were then clustered into subtopics by looking at the most frequent NPs they contain, with two exceptions. First, superstrings of the topic were favored as subtopics in order to discover complex nominals in which the topic appears. Secondly, non-reduced forms of personal names were required as subtopics, even if a reduced form was more frequent.

Similar heuristics were used to order paragraphs within the subtopic sections of the topic report as in the overview section.

Additional constraints were applied to stay within the boundaries of fair use of potentially copyrighted material, limiting the amount of contiguous text from any one source.

Topic reports were set to be refreshed by the system five days after they were generated in order to reflect any new developments.

In an evaluation of 642 paragraphs, 88.8% were relevant to the topic; 83.4% relevant to the topic as categorized. For images, 85.5% of 83 images were relevant, using a revised algorithm, not the live system. Of 1861 subtopic paragraphs, 88.5% of paragraphs were relevant to the assigned topic and subtopic.

9 Discussion

Of the over 30K topical reports generated by Retriever thus far, some of the reports generated turned out surprisingly well, while many turned out poorly. In general, since we paid no attention to temporal ordering of paragraphs, topics that were highly temporal did poorly, since we would typically arrange paragraphs with no regard for event precedence.

There are many things that remained to be done with Retriever, including extracting paragraphs from non-HTML documents, auto-hyperlinking topics within Retriever pages (as in Wikipedia), finding more up-to-date sources for categorization, and verticalizing Retriever page generation for different types of topics (e.g. treating movies differently than people and both differently than diseases). Unfortunately, the project was essentially discontinued in February, 2006.

10 Related Work

Although there have been previous systems that learned to identify and summarize web documents on a particular topic (Allen et al, 1996) without attempting to fuse them into a narrative structure, we are not aware of any project that attempts to generate coherent, narrative topical summaries by *paragraph* extraction and ordering. Much recent work focuses on multi-article summarization of news by *sentence* extraction and ordering (see for example, Columbia's well-known Newsblaster project and Michigan's NewsInEssence project). The latest DUC competition similarly emphasized sentence-level fusion of multi-document summaries from news text (DUC, 2005). One exception is the ArteQuaKt project (Kim et al, 2002), a prototype system for generating artist biographies from extracted passages and facts found on the Web aimed at different levels of readers (e.g. grade school versus university students). The Artequakt system was to use extracted text both as found and as generated from facts in a logical representation. It is not clear how far the ArteQuaKt project progressed.

Less legitimately, more and more "spam blogs" repackage snippets from search results or in other ways appropriate text from original sources into pages they populate with pay-per-click advertising. Retriever differs from such schemes in filtering out low value content and by making obscure sources visible.

References

Allen, Brad et al. 1996. WebCompass: an agent-based metasearch and metadata discovery tool for the Web. *SIGIR '96*.

Asher,Nicholas. 2004. Discourse Topic, *Theoretical Linguistics*. 30:2-3

DUC. 2005 DUC Workshop. Vancouver, BC

Kim, Sanghee et al. 2002. Artequakt: Generating Talored Biographies from Automatically Annotated Fragments from the Web. In *Proceedings of Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM'02)*. pp. 1-6, Lyon, France.

Liu, Bing, et al. 2003. Mining Topic-Specific Concepts and Definitions on the Web. Proceedings of the Twelfth International World Wide Web Conference (WWW-2003),

Shen, Dou et al, Q2C@UST: Our Winning Solution to Query Classification in KDDCUP 2005. *ACM KDD Explorations*. Vol 7, no. 2. December 2005.

Improved Affinity Graph Based Multi-Document Summarization

Xiaojun Wan, Jianwu Yang
Institute of Computer Science and Technology, Peking University
Beijing 100871, China
{wanxiaojun, yangjianwu}@icst.pku.edu.cn

Abstract

This paper describes an affinity graph based approach to multi-document summarization. We incorporate a diffusion process to acquire semantic relationships between sentences, and then compute information richness of sentences by a graph rank algorithm on differentiated intra-document links and inter-document links between sentences. A greedy algorithm is employed to impose diversity penalty on sentences and the sentences with both high information richness and high information novelty are chosen into the summary. Experimental results on task 2 of DUC 2002 and task 2 of DUC 2004 demonstrate that the proposed approach outperforms existing state-of-the-art systems.

1 Introduction

Automated multi-document summarization has drawn much attention in recent years. Multi-document summary is usually used to provide concise topic description about a cluster of documents and facilitate the users to browse the document cluster. A particular challenge for multi-document summarization is that the information stored in different documents inevitably overlaps with each other, and hence we need effective summarization methods to merge information stored in different documents, and if possible, contrast their differences.

A variety of multi-document summarization methods have been developed recently. In this study, we focus on extractive summarization, which involves assigning saliency scores to some units (e.g. sentences, paragraphs) of the documents and extracting the sentences with highest scores. MEAD is an implementation of the centroid-based method (Radev et al., 2004) that scores sentences based on sentence-level and inter-sentence features, including cluster centroids, position, TF*IDF, etc. NeATS (Lin and Hovy, 2002) selects important content using sentence position, term frequency, topic signature and term clustering, and then uses MMR (Goldstein et al., 1999) to remove redundancy. XDoX (Hardy et al., 1998) identifies the most salient themes within the set by passage clustering and then composes an extraction summary, which reflects these main themes. Harabagiu and Lacatusu (2005) investigate different topic representations and extraction methods.

Graph-based methods have been proposed to rank sentences or passages. Websumm (Mani and Bloedorn, 2000) uses a graph-connectivity model and operates under the assumption that nodes which are connected to many other nodes are likely to carry salient information. LexPageRank (Erkan and Radev, 2004) is an approach for computing sentence importance based on the concept of eigenvector centrality. Mihalcea and Tarau (2005) also propose similar algorithms based on PageRank and HITS to compute sentence importance for document summarization.

In this study, we extend the above graph-based works by proposing an integrated framework for considering both information richness and information novelty of a sentence based on sentence affinity graph. First, a diffusion process is imposed on sentence affinity graph in order to make the affinity graph reflect true semantic relationships between sentences. Second, intra-document links and inter-document links between sentences are differentiated to attach more importance to inter-document links for sentence information richness computation. Lastly, a diversity penalty process is imposed on sentences to penalize redundant sentences. Experiments on DUC 2002 and DUC 2004 data are performed and we obtain encouraging results and conclusions.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 181–184,
New York, June 2006. ©2006 Association for Computational Linguistics

2 The Affinity Graph Based Approach

The proposed affinity graph based summarization method consists of three steps: (1) an affinity graph is built to reflect the semantic relationship between sentences in the document set; (2) information richness of each sentence is computed based on the affinity graph; (3) based on the affinity graph and the information richness scores, diversity penalty is imposed to sentences and the affinity rank score for each sentence is obtained to reflect both information richness and information novelty of the sentence. The sentences with high affinity rank scores are chosen to produce the summary.

2.1 Affinity Graph Building

Given a sentence collection S={s_i | 1≤i≤n}, the affinity weight aff(s_i, s_j) between a sentence pair of s_i and s_j is calculated using the cosine measure. The weight associated with term t is calculated with the tf_t*isf_t formula, where tf_t is the frequency of term t in the corresponding sentence and isf_t is the inverse sentence frequency of term t, i.e. 1+log(N/n_t), where N is the total number of sentences and n_t is the number of sentences containing term t. If sentences are considered as nodes, the sentence collection can be modeled as an undirected graph by generating the link between two sentences if their affinity weight exceeds 0, i.e. an undirected link between s_i and s_j (i≠j) with affinity weight aff(s_i,s_j) is constructed if aff(s_i,s_j)>0; otherwise no link is constructed. Thus, we construct an undirected graph G reflecting the semantic relationship between sentences by their content similarity. The graph is called as Affinity Graph. We use an adjacency (affinity) matrix **M** to describe the affinity graph with each entry corresponding to the weight of a link in the graph. **M** = $(M_{i,j})_{n\times n}$ is defined as follows:

$$M_{i,j} = \mathrm{aff}(s_i, s_j) \qquad (1)$$

Then **M** is normalized to make the sum of each row equal to 1. Note that we use the same notation to denote a matrix and its normalized matrix.

However, the affinity weight between two sentences in the affinity graph is currently computed simply based on their own content similarity and ignore the affinity diffusion process on the graph. Other than the direct link between two sentences, the possible paths with more than two steps between the sentences in the graph also convey more or less semantic relationship. In order to acquire the implicit semantic relationship between sentences, we apply a diffusion process（Kandola et al., 2002）on the graph to obtain a more appropriate affinity matrix. Though the number of possible paths between any two given nodes can grow exponentially, recent spectral graph theory (Kondor and Lafferty, 2002) shows that it is possible to compute the affinity between any two given nodes efficiently without examining all possible paths. The diffusion process on the graph is as follows:

$$\tilde{\mathbf{M}} = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbf{M}^t \qquad (2)$$

where γ(0<γ<1) is the decay factor set to 0.9. $\mathbf{M}^t$ is the t-th power of the initial affinity matrix **M** and the entry in it is given by

$$M^t_{i,j} = \sum_{\substack{u\in\{1,\ldots,n\}^t \\ u_1=i,\, u_t=j}} \prod_{\ell=1}^{t-1} M_{u_\ell, u_{\ell+1}} \qquad (3)$$

that is the sum of the products of the weights over all paths of length t that start at node i and finish at node j in the graph on the examples. If the entries satisfy that they are all positive and for each node the sum of the connections is 1, we can view the entry as the probability that a random walk beginning at node i reaches node j after t steps. The matrix $\tilde{\mathbf{M}}$ is normalized to make the sum of each row equal to 1. t is limited to 5 in this study.

2.2 Information Richness Computation

The computation of information richness of sentences is based on the following three intuitions: 1) the more neighbors a sentence has, the more informative it is; 2) the more informative a sentence's neighbors are, the more informative it is; 3) the more heavily a sentence is linked with other informative sentences, the more informative it is. Based on the above intuitions, the information richness score InfoRich(s_i) for a sentence s_i can be deduced from those of all other sentences linked with it and it can be formulated in a recursive form as follows:

$$\mathrm{InfoRich}(s_i) = d \cdot \sum_{\text{all } j\neq i} \mathrm{InfoRich}(s_j) \cdot \tilde{M}_{j,i} + \frac{(1-d)}{n} \qquad (4)$$

And the matrix form is:

$$\vec{\lambda} = d\tilde{\mathbf{M}}^T \vec{\lambda} + \frac{(1-d)}{n}\vec{e} \qquad (5)$$

where $\vec{\lambda} = [\mathrm{InfoRich}(s_i)]_{n\times 1}$ is the eigenvector of $\widetilde{\mathbf{M}}^{T}$. $\vec{e}$ is a unit vector with all elements equaling to 1. d is the damping factor set to 0.85.

Note that given a link between a sentence pair of s_i and s_j, if s_i and s_j comes from the same document, the link is an intra-document link; and if s_i and s_j comes from different documents, the link is an inter-document link. We believe that inter-document links are more important than intra-document links for information richness computation. Different weights are assigned to intra-document links and inter-document links respectively, and the new affinity matrix is:

$$\hat{\mathbf{M}} = \alpha\widetilde{\mathbf{M}}_{\mathrm{intra}} + \beta\widetilde{\mathbf{M}}_{\mathrm{inter}} \qquad (6)$$

where $\widetilde{\mathbf{M}}_{\mathrm{intra}}$ is the affinity matrix containing only the intra-document links (the entries of inter-document links are set to 0) and $\widetilde{\mathbf{M}}_{\mathrm{inter}}$ is the affinity matrix containing only the inter-document links (the entries of intra-document links are set to 0). α, β are weighting parameters and we let 0≤α, β≤1. The matrix is normalized and now the matrix $\widetilde{\mathbf{M}}$ is replaced by $\hat{\mathbf{M}}$ in Equations (4) and (5).

2.3 Diversity Penalty Imposition

Based on the affinity graph and obtained information richness scores, a greedy algorithm is applied to impose the diversity penalty and compute the final affinity rank scores of sentences as follows:

1. Initialize two sets A=Ø, B={s_i | i=1,2,…,n}, and each sentence's affinity rank score is initialized to its information richness score, i.e. ARScore(s_i) = InfoRich(s_i), i=1,2,…n.
2. Sort the sentences in B by their current affinity rank scores in descending order.
3. Suppose s_i is the highest ranked sentence, i.e. the first sentence in the ranked list. Move sentence s_i from B to A, and then a diversity penalty is imposed to the affinity rank score of each sentence linked with s_i as follows:
   For each sentence s_j in B, we have
   $$\mathrm{ARScore}(s_j) = \mathrm{ARScore}(s_j) - \omega\cdot\widetilde{M}_{j,i}\cdot\mathrm{InfoRich}(s_i) \qquad (7)$$
   where ω>0 is the penalty degree factor. The larger ω is, the greater penalty is imposed to the affinity rank score. If ω=0, no diversity penalty is imposed at all.
4. Go to step 2 and iterate until B= Ø or the iteration count reaches a predefined maximum number.

After the affinity rank scores are obtained for all sentences, the sentences with highest affinity rank scores are chosen to produce the summary according to the summary length limit.

3 Experiments and Results

We compare our system with top 3 performing systems and two baseline systems on task 2 of DUC 2002 and task 4 of DUC 2004 respectively. ROUGE (Lin and Hovy, 2003) metrics is used for evaluation[1] and we mainly concern about ROUGE-1. The parameters of our system are tuned on DUC 2001 as follows: ω=7, α=0.3 and β=1.

We can see from the tables that our system outperforms the top performing systems and baseline systems on both DUC 2002 and DUC 2004 tasks over all three metrics. The performance improvement achieved by our system results from three factors: diversity penalty imposition, intra-document and inter-document link differentiation and diffusion process incorporation. The ROUGE-1 contributions of the above three factors are 0.02200, 0.00268 and 0.00043 respectively.

System	ROUGE-1	ROUGE-2	ROUGE-W
Our System	**0.38125**	**0.08196**	**0.12390**
S26	0.35151	0.07642	0.11448
S19	0.34504	0.07936	0.11332
S28	0.34355	0.07521	0.10956
Coverage Baseline	0.32894	0.07148	0.10847
Lead Baseline	0.28684	0.05283	0.09525

Table 1. System comparison on task 2 of DUC 2002

System	ROUGE-1	ROUGE-2	ROUGE-W
Our System	**0.41102**	**0.09738**	**0.12560**
S65	0.38232	0.09219	0.11528
S104	0.37436	0.08544	0.11305
S35	0.37427	0.08364	0.11561
Coverage Baseline	0.34882	0.07189	0.10622
Lead Baseline	0.32420	0.06409	0.09905

Table 2. System comparison on task 2 of DUC 2004

Figures 1-4 show the influence of the parameters in our system. Note that α: β denotes the real values α and β are set to. "w/ diffusion" is the system with the diffusion process (our system) and "w/o diffusion" is the system without the diffusion proc-

[1] We use ROUGEeval-1.4.2 with "-l" or "-b" option for truncating longer summaries, and "-m" option for word stemming.

ess. The observations demonstrate that “w/ diffusion” performs better than “w/o diffusion” for most parameter settings. Meanwhile, “w/ diffusion” is more robust than “w/o diffusion” because the ROUGE-1 value of “w/ diffusion” changes less when the parameter values vary. Note that in Figures 3 and 4 the performance decreases sharply with the decrease of the weight β of inter-document links and it is the worst case when inter-document links are not taken into account (i.e. α:β=1:0), while if intra-document links are not taken into account (i.e. α:β=0:1), the performance is still good, which demonstrates the great importance of inter-document links.

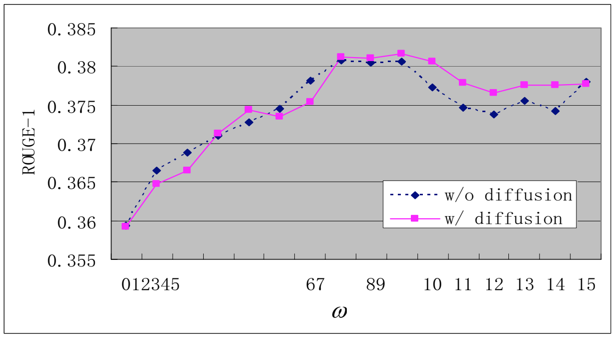


Figure 1. Penalty factor tuning on task 2 of DUC 2002

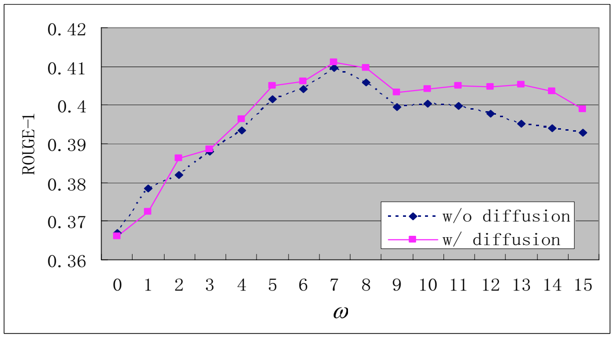


Figure 2. Penalty factor tuning on task 2 of DUC 2004

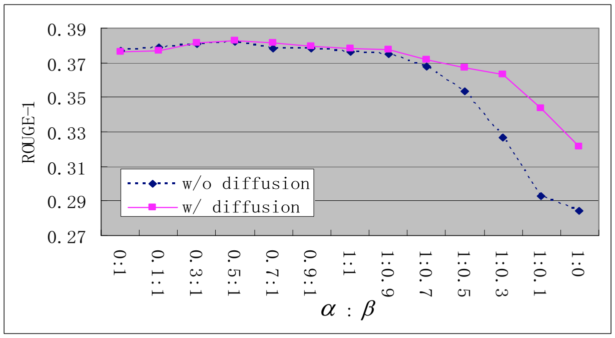


Figure3. Intra- & Inter-document link weight tuning on task 2 of DUC 2002

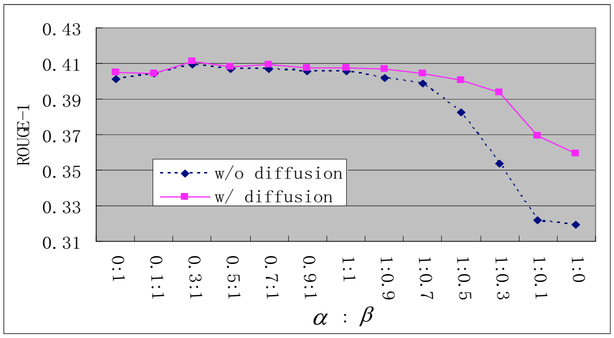


Figure 4. Intra- & Inter-document link weight tuning on task 2 of DUC 2004

References

G. Erkan and D. Radev. LexPageRank: prestige in multi-document text summarization. In Proceedings of EMNLP’04

J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. Proceedings of SIGIR-99.

S. Harabagiu and F. Lacatusu. Topic themes for multi-document summarization. In Proceedings of SIGIR’05, Salvador, Brazil, 202-209, 2005.

H. Hardy, N. Shimizu, T. Strzalkowski, L. Ting, G. B. Wise, and X. Zhang. Cross-document summarization by concept classification. In Proceedings of SIGIR’02, Tampere, Finland, 2002.

J. Kandola, J. Shawe-Taylor, N. Cristianini. Learning semantic similarity. In Proceedings of NIPS’2002.

K. Knight and D. Marcu. Summarization beyond sentence extraction: a probabilistic approach to sentence compression, Artificial Intelligence, 139(1), 2002.

R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In Proceedings of ICML’2002.

C.-Y. Lin and E.H. Hovy. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In Proceedings of HLT-NAACL 2003.

C.-Y. Lin and E.H. Hovy. From Single to Multi-document Summarization: A Prototype System and its Evaluation. In Proceedings of ACL-2002.

I. Mani and E. Bloedorn. Summarizing Similarities and Differences Among Related Documents. Information Retrieval, 1(1), 2000.

R. Mihalcea and P. Tarau. A language independent algorithm for single and multiple document summarization. In Proceedings of IJCNLP’2005.

D. R. Radev, H. Y. Jing, M. Stys and D. Tam. Centroid-based summarization of multiple documents. Information Processing and Management, 40: 919-938, 2004.

A Maximum Entropy Framework that Integrates Word Dependencies and Grammatical Relations for Reading Comprehension

Kui Xu[1,2] **and Helen Meng**[1]
[1]Human-Computer Communications Laboratory
Dept. of Systems Engineering and
Engineering Management
The Chinese University of Hong Kong
Hong Kong SAR, China
{kxu, hmmeng}@se.cuhk.edu.hk

Fuliang Weng[2]
[2]Research and Technology Center
Robert Bosch Corp.
Palo Alto, CA 94304, USA
Fuliang.weng@rtc.bosch.com

Abstract

Automatic reading comprehension (RC) systems can analyze a given passage and generate/extract answers in response to questions about the passage. The RC passages are often constrained in their lengths and the target answer sentence usually occurs very few times. In order to generate/extract a specific precise answer, this paper proposes the integration of two types of "deep" linguistic features, namely word dependencies and grammatical relations, in a maximum entropy (ME) framework to handle the RC task. The proposed approach achieves 44.7% and 73.2% HumSent accuracy on the Remedia and ChungHwa corpora respectively. This result is competitive with other results reported thus far.

1 Introduction

Automatic reading comprehension (RC) systems can analyze a given passage and generate/extract answers in response to questions about the passage. The RC passages are often constrained in their lengths and the target answer sentence usually occurs only once (or very few times). This differentiates the RC task from other tasks such as open-domain question answering (QA) in the Text Retrieval Conference (Light et al., 2001). In order to generate/extract a specific precise answer to a given question from a short passage, "deep" linguistic analysis of sentences in a passage is needed.

Previous efforts in RC often use the bag-of-words (BOW) approach as the baseline, which is further augmented with techniques such as shallow syntactic analysis, the use of named entities (NE) and pronoun references. For example, Hirschman et al. (1999) have augmented the BOW approach with stemming, NE recognition, NE filtering, semantic class identification and pronoun resolution to achieve 36% HumSent[1] accuracy in the Remedia test set. Based on these technologies, Riloff and Thelen (2000) improved the HumSent accuracy to 40% by applying a set of heuristic rules that assign handcrafted weights to matching words and NE. Charniak et al. (2000) used additional strategies for different question types to achieve 41%. An example strategy for *why* questions is that if the first word of the matching sentence is "this," "that," "these" or "those," the system should select the previous sentence as an answer. Light et al. (2001) also introduced an approach to estimate the performance upper bound of the BOW approach. When we apply the same approach to the Remedia test set, we obtained the upper bound of 48.3% HumSent accuracy. The state-of-art performance reached 42% with answer patterns derived from web (Du et al., 2005).

This paper investigates the possibility of enhancing RC performance by applying "deep" linguistic analysis for every sentence in the passage. We refer to the use of two types of features, namely word dependencies and grammatical relations, that

[1] If the system's answer sentence is identical to the corresponding human marked answer sentence, the question scores one point. Otherwise, the question scores no point. HumSent accuracy is the average score across all questions.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 185–188,
New York, June 2006. ©2006 Association for Computational Linguistics

are integrated in a maximum entropy framework. Word dependencies refer to the headword dependencies in lexicalized syntactic parse trees, together with part-of-speech (POS) information. Grammatical relations (GR) refer to linkages such as subject, object, modifier, etc. The ME framework has shown its effectiveness in solving QA tasks (Ittycheriah et al., 1994). In comparison with previous approaches mentioned earlier, the current approach involves richer syntactic information that cover longer-distance relationships.

2 Corpora

We used the Remedia corpus (Hirschman et al., 1999) and ChungHwa corpus (Xu and Meng, 2005) in our experiments. The Remedia corpus contains 55 training stories and 60 testing stories (about 20K words). Each story contains 20 sentences on average and is accompanied by five types of questions: *who, what, when, where* and *why*. The ChungHwa corpus contains 50 training stories and 50 test stories (about 18K words). Each story contains 9 sentences and is accompanied by four questions on average. Both the Remedia and ChungHwa corpora contain the annotation of NE, anaphor referents and answer sentences.

3 The Maximum Entropy Framework

Suppose a story S contains n sentences, $C_0, \ldots, C_n$, the objective of an RC system can be described as:

$$A = \arg\max_{C_i \in S} P(C_i \textit{ answers } Q|Q). \quad (1)$$

Let "x" be the question (Q) and "y" be the answer sentence C_i that answers "x". Equation 1 can be computed by the ME method (Zhou et al., 2003):

$$p(y|x) = \frac{1}{Z(x)} \exp^{\sum_j \lambda_j f_j(x,y)}, \quad (2)$$

where $Z(x) = \sum_y \exp^{\sum_j \lambda_j f_j(x,y)}$ is a normalization factor, $f_j(x,y)$ is the indicator function for feature f_j; f_j occurs in the context x, λ_j is the weight of f_j. For a given question Q, the C_i with the highest probability is selected. If multiple sentences have the maximum probability, the one that occurs the earliest in the passage is returned. We used the selective gain computation (SGC) algorithm (Zhou et al., 2003) to select features and estimate parameters for its fast performance.

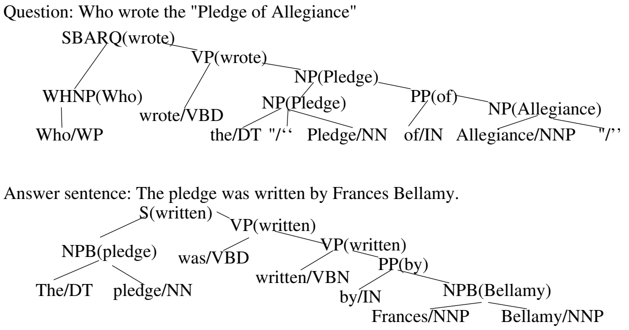


Figure 1. The lexicalized syntactic parse trees of a question and a candidate answer sentence.

4 Features Used in the "Deep" Linguistic Analysis

A feature in the ME approach typically has binary values: $f_j(x,y) = 1$ if the feature j occurs; otherwise $f_j(x,y) = 0$. This section describes two types of "deep" linguistic features to be integrated in the ME framework in two subsections.

4.1 POS Tags of Matching Words and Dependencies

Consider the following question Q and sentence C,

Q: *Who wrote the "Pledge of Allegiance"*
C: *The pledge was written by Frances Bellamy.*

The set of words and POS tags[2] are:

Q: {*write/VB, pledge/NN, allegiance/NNP*}
C: {*write/VB, pledge/NN, by/IN, Frances/NNP, Bellamy/NNP*}.

Two matching words between Q and C (i.e. "*write*" and "*pledge*") activate two POS tag features:

$$f_{VB}(x,y)\text{=1 and } f_{NN}(x,y)\text{=1}.$$

We extracted dependencies from lexicalized syntactic parse trees, which can be obtained according to the head-rules in (Collins, 1999) (e.g. see Figure 1). In a lexicalized syntactic parse tree, a dependency can be defined as:

$$< hc \rightarrow hp > \text{ or } < hr \rightarrow TOP >,$$

where hc is the headword of the child node, hp is the headword of the parent node ($hc \neq hp$), hr is the headword of the root node. Sample

[2] We used the MXPOST toolkit downloaded from *ftp://ftp.cis.upenn.edu/pub/adwait/jmx/* to generate POS tags. Stop words including *who, what, when, where, why, be, the, a, an*, and *of* are removed in all questions and story sentences. All plural noun POS tags are replaced by their single forms (e.g. NNS→NN); all verb POS tags are replaced by their base forms (e.g. VBN→VB) due to stemming.

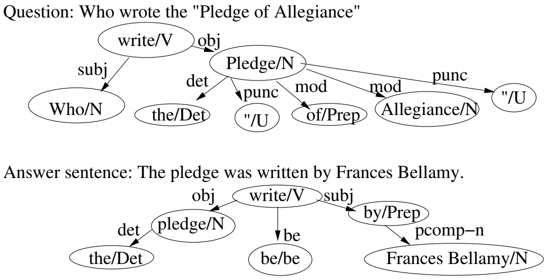


Figure 2. The dependency trees produced by MINIPAR for a question and a candidate answer sentence.

dependencies in C (see Figure 1) are:

<write→TOP> and <pledge→write>.

The dependency features are represented by the combined POS tags of the modifiers and headwords of (identical) matching dependencies[3]. A matching dependency between Q and C, *<pledge→write>* activates a dependency feature: $f_{NN-VB}(x,y)$=1. In total, we obtained 169 and 180 word dependency features from the Remedia and ChungHwa training sets respectively.

4.2 Matching Grammatical Relationships (GR)

We extracted grammatical relationships from the dependency trees produced by MINIPAR (Lin, 1998), which covers 79% of the dependency relationships in the SUSANNE corpus with 89% precision[4]. IN a MINIPAR dependency relationship:

$(word1$ *CATE1:RELATION:CATE2* $word2)$, CATE1 and CATE2 represent such grammatical categories as nouns, verbs, adjectives, etc.; RELATION represents the grammatical relationships such as subject, objects, modifiers, etc.[5] Figure 2 shows dependency trees of Q and C produced by MINIPAR. Sample grammatical relationships in C are *pledge N:det:Det the,* and *write V:by-subj:Prep by.* GR features are extracted from identical matching relationships between questions and candidate sentences. The only identical matching relationship between Q and C, "*write V:obj:N pledge*" activates a grammatical relationship feature: $f_{obj}(x,y)$=1. In total, we extracted 44 and 45 GR features from the Remedia and ChungHwa training sets respectively.

5 Experimental Results

We selected the features used in Quarc (Riloff and Thelen, 2000) to establish the reference performance level. In our experiments, the 24 rules in Quarc are transferred[6] to ME features:

"If contains(Q,{*start, begin*}) and contains(S,{*start, begin, since, year*}) Then Score(S)+=20" → $f_j(x,y) = 1$ (0< j <25) if Q is a *when* question that contains *"start"* or *"begin"* and C contains *"start," "begin," "since"* or *"year"*; $f_j(x,y) = 0$ otherwise.

In addition to the Quarc features, we resolved five pronouns (*he, him, his, she* and *her*) in the stories based on the annotation in the corpora. The result of using Quarc features in the ME framework is 38.3% HumSent accuracy on the Remedia test set. This is lower than the result (40%) obtained by our re-implementation of Quarc that uses handcrafted scores. A possible explanation is that handcrafted scores are more reliable than ME, since humans can generalize the score even for sparse data. Therefore, we refined our reference performance level by combining the ME models (MEM) and handcrafted models (HCM). Suppose the score of a question-answer pair is $score(Q,C_i)$, the conditional probability that C_i answers Q in HCM is:

$$HCM(Q,C_i) = P(C_i \textit{ answers } Q|Q) = \frac{score(Q,C_i)}{\Sigma_{j\leq n} score(Q,C_j)}.$$

We combined the probabilities from MEM and HCM in the following manner:

$$score'(Q,C_i) = \alpha MEM(Q,C_i) + (1-\alpha)HCM(Q,C_i).$$

To obtain the optimal α, we partitioned the training set into four bins. The ME models are trained on three different bins; the optimal α is determined on the other bins. By trying different bins combinations and different α such that $0 < \alpha < 1$ with interval 0.1, we obtained the average optimal $\alpha = 0.15$ and 0.9 from the Remedia and ChungHwa training sets respectively[7]. Our baseline used the combined ME models and handcrafted models to achieve 40.3% and 70.6% HumSent accuracy in the Remedia and ChungHwa test sets respectively.

We set up our experiments such that the linguistic features are applied incrementally - (i) First , we use only POS tags of matching words among questions

[3] We extracted dependencies from parse trees generated by Collins' parser (Collins, 1999).

[4] MINIPAR outputs GR directly, while Collins' parser gives better result for dependencies.

[5] Refer to the *readme* file of MINIPAR downloaded from *http://www.cs.ualberta.ca/ lindek/minipar.htm*

[6] The features in (Charniak et al., 2000) and (Du et al., 2005) could have been included similarly if they were available.

[7] HCM are tuned by hand on Remedia, thus a bigger weight, 0.85 represents their reliability. For ChungHwa, a weight, 0.1 means that HCM are less reliable.

and candidate answer sentences. (ii) Then we add POS tags of the matching dependencies. (iii) We apply only GR features from MINIPAR. (iv) All features are used. These four feature sets are denoted as “+wp,” “+wp+dp,” “+mini” and “+wp+dp+mini” respectively. The results are shown in Figure 3 for the Remedia and ChungHwa test sets.

With the significance level 0.05, the pairwise t-test (for every question) to the statistical significance of the improvements shows that the p-value is 0.009 and 0.025 for the Remedia and ChungHwa test sets respectively. The “deep” syntactic features significantly improve the performance over the baseline system on the Remedia and ChungHwa test sets[8].

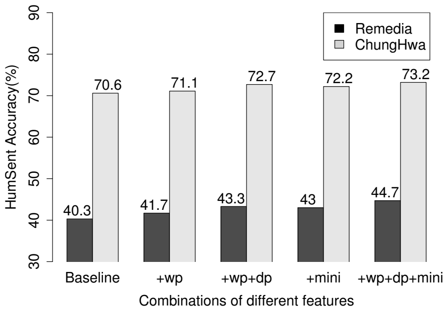


Figure 3. Baseline and proposed feature results on the Remedia and ChungHwa test sets.

6 Conclusions

This paper proposes the integration of two types of “deep” linguistic features, namely word dependencies and grammatical relations, in a ME framework to handle the RC task. Our system leverages linguistic information such as POS, word dependencies and grammatical relationships in order to extract the appropriate answer sentence for a given question from all available sentences in the passage. Our system achieves 44.7% and 73.2% HumSent accuracy on the Remedia and ChungHwa test sets respectively. This shows a statistically significant improvement over the reference performance levels, 40.3% and 70.6% on the same test sets.

Acknowledgements

This work is done during the first author’s internship at RTC Bosch Corp. The work is also affiliated with the CUHK Shun Hing Institute of Advanced Engineering and partially supported by CUHK4237/03E from RGC of HKSAR Government.

References

Dekang Lin. 1998. *Dependency-based Evaluation of MINIPAR*. Workshop on the Evaluation of Parsing Systems 1998.

Ellen Riloff and Michael Thelen. 2000. *A Rule-based Question Answering System for Reading Comprehension Test*. ANLP/NAACL-2000 Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems.

Eugene Charniak, Yasemin Altun, Rofrigo D. Braz, Benjamin Garrett, Margaret Kosmala, Tomer Moscovich, Lixin Pang, Changhee Pyo, Ye Sun, Wei Wy, Zhongfa Yang, Shawn Zeller, and Lisa Zorn. 2000. *Reading Comprehension Programs In a Statistical-Language-Processing Class*. ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems.

Kui Xu and Helen Meng. 2005. *Design and Development of a Bilingual Reading Comprehension Corpus*. International Journal of Computational Linguistics & Chinese Language Processing, Vol. 10, No. 2.

Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. *Deep Read: A Reading Comprehension System*. Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics.

Marc Light, Gideon S. Mann, Ellen Riloff, and Eric Breck. 2001. *Analyses for Elucidating Current Question Answering Technology*. Journal of Natural Language Engineering, No. 4 Vol. 7.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

Abraham Ittycheriah, Martin Franz, Wei-Jing Zhu and Adwait Ratnaparkhi. 2001. *Question Answering Using Maximum-Entropy Components*. Proceedings of NAACL 2001.

Yaqian Zhou, Fuliang Weng, Lide Wu, Hauke Schmidt. 2003. *A Fast Algorithm for Feature Selection in Conditional Maximum Entropy Modeling*. Proceedings of EMNLP 2003.

Yongping Du, Helen Meng, Xuanjing Huang, Lide Wu. 2005. *The Use of Metadata, Web-derived Answer Patterns and Passage Context to Improve Reading Comprehension Performance*. Proceedings of HLT/EMNLP 2005.

[8] Our previous work about developing the ChungHwa corpus (Xu and Meng, 2005) shows that most errors can only be solved by reasoning with domain ontologies and world knowledge.

BioEx: A Novel User-Interface that Accesses Images from Abstract Sentences

Hong Yu
Department of Biomedical Informatics
Columbia University
New York, NY 10032
Hy52@columbia.edu

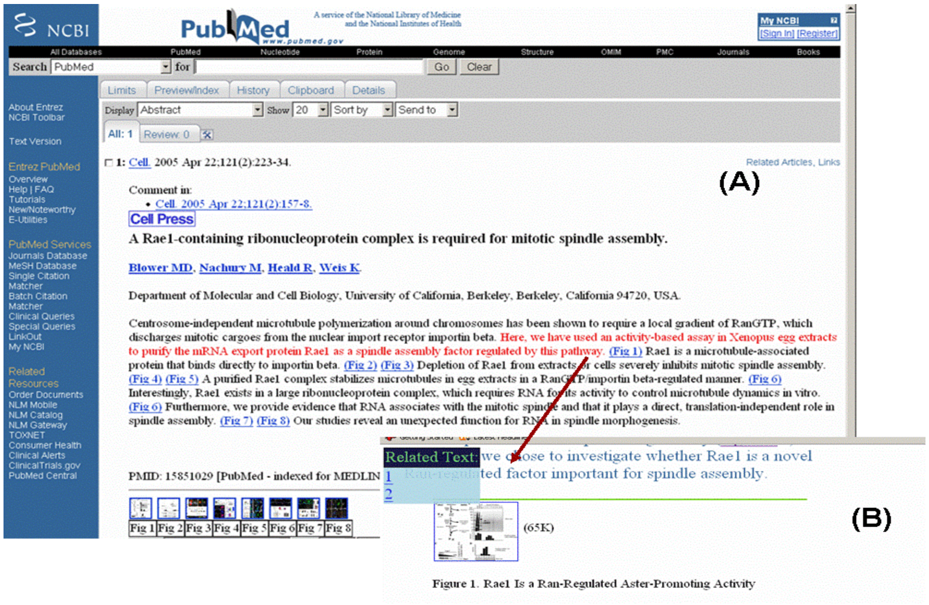
Minsuk Lee
Department of Biomedical Informatics
Columbia University
New York, NY 10032
minsuk.lee@gmail.com

Abstract

Images (i.e., figures or tables) are important experimental results that are typically reported in bioscience full-text articles. Biologists need to access the images to validate research facts and to formulate or to test novel research hypotheses. We designed, evaluated, and implemented a novel user-interface, BioEx, that allows biologists to access images that appear in a full-text article directly from the abstract of the article.

1 Introduction

The rapid growth of full-text electronic publications in bioscience has made it necessary to create information systems that allow biologists to navigate and search efficiently among them. Images are usually important experimental results that are typically reported in full-text bioscience articles. An image is worth a thousand words. Biologists need to access image data to validate research facts and to formulate or to test novel research hypotheses. Additionally, full-text articles are frequently long and typically incorporate multiple images. For example, we have found an average of 5.2 images per biological article in the journal *Proceedings of the National Academy of Sciences* (PNAS). Biologists need to spend significant amount of time to read the full-text articles in order to access specific images.



□ 1: Cell. 2005 Apr 22;121(2):223-34.

Comment in:
- Cell. 2005 Apr 22;121(2):157-8.

A Rae1-containing ribonucleoprotein complex is required for mitotic spindle assembly.

Blower MD, Nachury M, Heald R, Weis K.

Department of Molecular and Cell Biology, University of California, Berkeley, Berkeley, California 94720, USA.

Centrosome-independent microtubule polymerization around chromosomes has been shown to require a local gradient of RanGTP, which discharges mitotic cargoes from the nuclear import receptor importin beta. Here, we have used an activity-based assay in Xenopus egg extracts to purify the mRNA export protein Rae1 as a spindle assembly factor regulated by this pathway. (Fig 1) Rae1 is a microtubule-associated protein that binds directly to importin beta. (Fig 2) (Fig 3) Depletion of Rae1 from extracts or cells severely inhibits mitotic spindle assembly. (Fig 4) (Fig 5) A purified Rae1 complex stabilizes microtubules in egg extracts in a RanGTP/importin beta-regulated manner. (Fig 6) Interestingly, Rae1 exists in a large ribonucleoprotein complex, which requires RNA for its activity to control microtubule dynamics in vitro. (Fig 6) Furthermore, we provide evidence that RNA associates with the mitotic spindle and that it plays a direct, translation-independent role in spindle assembly. (Fig 7) (Fig 8) Our studies reveal an unexpected function for RNA in spindle morphogenesis.

PMID: 15851029 [PubMed - indexed for MEDLIN

we chose to investigate whether Rae1 is a novel Ran-regulated factor important for spindle assembly.

Figure 1. Rae1 Is a Ran-Regulated Aster-Promoting Activity

Figure 1. BioEx user-interface (as shown in A) is built upon the PubMed user-interface. Images are shown as thumbnails at the bottom of a PubMed abstract. Images include both Figure and Table. When a mouse (as shown as a hand in A) moves to "Fig x", it shows the associated abstract sentence(s) that link to the original figure that appears in the full-text articles. For example, "Fig 1" links to image B. "Related Text" provides links to other associated texts that correspond to the image besides its image caption.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 189–192,
New York, June 2006.

In order to facilitate biologists' access to images, we designed, evaluated, and implemented a novel user-interface, BioEx, that allows biologists to access images that appear in a full-text article directly from the abstract of the article. In the following, we will describe the BioEx user-interface, evaluation, and the implementation.

2. Data Collection

We hypothesize that images reported in a full-text article can be summarized by sentences in the abstract. To test this hypothesis, we randomly selected a total of 329 biological articles that are recently published in leading journals *Cell* (104), *EMBO* (72), *Journal of Biological Chemistry* (92), and *Proceedings of the National Academy of Sciences (PNAS)* (61). For each article, we e-mailed the corresponding author and invited him or her to identify abstract sentences that summarize image content in that article. In order to eliminate the errors that may be introduced by sentence boundary ambiguity, we manually segmented the abstracts into sentences and sent the sentences as the email attachments.

A total of 119 biologists from 19 countries participated voluntarily the annotation to identify abstract sentences that summarize figures or tables from 114 articles (39 *Cells*, 29 *EMBO*, 30 *Journal of Biological Chemistry*, and 16 *PNAS*), a collection that is 34.7% of the total articles we requested. The responding biologists included the corresponding authors to whom we had sent emails, as well as the first authors of the articles to whom the corresponding authors had forwarded our emails. None of the biologists or authors were compensated.

This collection of 114 full-text articles incorporates 742 images and 826 abstract sentences. The average number of images per document is 6.5±1.5 and the average number of sentences per abstract is 7.2±1.9. Our data show that 87.9% images correspond to abstract sentences and 66.5% of the abstract sentences correspond to images. The data empirically validate our hypothesis that image content can be summarized by abstract sentences. Since an abstract is a summary of a full-text article, our results also empirically validate that images are important elements in full-text articles. This collection of 114 annotated articles was then used as the corpus to evaluate automatic mapping of abstract sentences to images using the natural language processing approaches described in Section 4.

3. BioEx User-Interface Evaluation

In order to evaluate whether biologists would prefer to accessing images from abstract sentence links, we designed BioEx (Figure 1) and two other baseline user-interfaces. BioEx is built upon the PubMed user-interface except that images can be accessed by the abstract sentences. We chose the PubMed user-interface because it has more than 70 million hits a month and represents the most familiar user-interface to biologists. Other information systems have also adapted the PubMed user-interface for similar reasons (Smalheiser and Swanson 1998; Hearst 2003). The two other baseline user-interfaces were the original PubMed user-interface and a modified version of the SummaryPlus user-interface, in which the images are listed as disjointed thumbnails rather than related by abstract sentences.

We asked the 119 biologists who linked sentences to images in their publications to assign a label to each of the three user-interfaces to be "My favorite", "My second favorite", or "My least favorite". We designed the evaluation so that a user-interface's label is independent of the choices of the other two user-interfaces.

A total of 41 or 34.5% of the biologists completed the evaluation in which 36 or 87.8% of the total 41 biologists judged BioEx as "My favorite". One biologist judged all three user-interfaces to be "My favorite". Five other biologists considered SummaryPlus as "My favorite", two of whom (or 4.9% of the total 41 biologists) judged BioEx to be "My least favorite".

4. Linking Abstract Sentences to Images

We have explored hierarchical clustering algorithms to cluster abstract sentences and image captions based on lexical similarities.
Hierarchical clustering algorithms are well-established algorithms that are widely used in

many other research areas including biological sequence alignment (Corpet 1988), gene expression analyses (Herrero et al. 2001), and topic detection (Lee et al. 2006). The algorithm starts with a set of text (i.e., abstract sentences or image captions). Each sentence or image caption represents a document that needs to be clustered. The algorithm identifies pair-wise document similarity based on the TF*IDF weighted cosine similarity. It then merges the two documents with the highest similarity into one cluster. It then re-evaluates pairs of documents/clusters; two clusters can be merged if the average similarity across all pairs of documents within the two clusters exceeds a predefined threshold. In presence of multiple clusters that can be merged at any time, the pair of clusters with the highest similarity is always preferred.

In our application, if abstract sentences belong to the same cluster that includes images captions, the abstract sentences summarize the image content of the corresponded images. The clustering model is advantageous over other models in that the flexibility of clustering methods allows "many-to-many" mappings. That is a sentence in the abstract can be mapped to zero, one or more than one images and an image can be mapped to zero, one or more than one abstract sentences.

We explored different learning features, weights and clustering algorithms to link abstract sentences to images. We applied the TF*IDF weighted cosine similarity for document clustering. We treat each sentence or image caption as a "document" and the features are bag-of-words.

We tested three different methods to obtain the IDF value for each word feature: 1) ***IDF(abstract+caption):*** the IDF values were calculated from the pool of abstract sentences and image captions; 2) ***IDF(full-text):*** the IDF values were calculated from all sentences in the full-text article; and 3) ***IDF(abstract)::IDF(caption):*** two sets of IDF values were obtained. For word features that appear in abstracts, the IDF values were calculated from the abstract sentences. For words that appear in image captions, the IDF values were calculated from the image captions.

The positions of abstract sentences or images are important. The chance that two abstract sentences link to an image decreases when the distance between two abstract sentences increases. For example, two consecutive abstract sentences have a higher probability to link to one image than two abstract sentences that are far apart. Two consecutive images have a higher chance to link to the same abstract sentence than two images that are separated by many other images. Additionally, sentence positions in an abstract seem to correspond to image positions. For example, the first sentences in an abstract have higher probabilities than the last sentences to link to the first image.

To integrate such "neighboring effect" into our existing hierarchical clustering algorithms, we modified the TF*IDF weighted cosine similarity. The TF*IDF weighted cosine similarity for a pair of documents *i* and *j* is *Sim(i,j)*, and the final similarity metric *W(i,j)* is:

$$W(i,j) = Sim(i,j) * (1 - abs(P_i / T_i - P_j / T_j))$$

1. If *i* and *j* are both abstract sentences, $T_i = T_j =$ *total number of abstract sentences;* and P_i and P_j represents the positions of sentences *i* and *j* in the abstract.

2. If *i* and *j* are both image captions, $T_i = T_j =$ *total number of images that appear in a full-text article;* and P_i and P_j represents the positions of images *i* and *j* in the full-text article.

3. If *i* and *j* are an abstract sentence and an image caption, respectively, $T_i =$ *total number of abstract sentences* and $T_j =$ *total number of images that appear in a full-text article;* and P_i and P_j represent the positions of abstract sentence *i* and image *j*.

Finally, we explored three clustering strategies; namely, *per-image, per-abstract sentence,* and *mix.*

The **Per-image** strategy clusters each image caption with all abstract sentences. The image is assigned to (an) abstract sentence(s) if it belongs to the same cluster. This method values features in abstract sentences more than image captions because the decision that an image belongs to (a) sentence(s) depends upon the features from all abstract sentences and the examined image caption. The features from other image captions do not play a role in the clustering methodology.

The **Per-abstract-sentence** strategy takes each abstract sentence and clusters it with all image captions that appear in a full-text article. Images are assigned to the sentence if they belong to the same cluster. This method values features in image captions higher than the features in abstract sentences because the decision that an abstract sentence belongs to image(s) depends upon the features from the image captions and the examined abstract sentence. Similar to per-image clustering, the features from other abstract sentences do not play a role in the clustering methodology.

The **Mix** strategy clusters all image captions with all abstract sentences. This method treats features in abstract sentences and image captions equally.

5. Results and Conclusions

Figures 2 - 4 show the results from three different combinations of features and algorithms with varied TF*IDF thresholds. The default parameters for all these experiments were “per image”, “bag-of-words”, and “without neighboring weight”.

Figure 2 shows that the “global” IDFs, or the IDFs obtained from the full-text article, have a much lower performance than “local” IDFs, or IDFs calculated from the abstract sentences and image captions. Figure 3 shows that **Per-image** out-performs the other two strategies. The results suggest that features in abstract sentences are more useful than features that reside within captions for the task of clustering. Figure 4 shows that the “neighboring weighted” approach offers significant enhancement over the TF*IDF weighted approach. When the recall is 33%, the precision of “neighboring weighted” approach increases to 72% from the original 38%, which corresponds to a 34% increase. The results strongly indicate the importance of the “neighboring effect” or positions of additional features. When the precision is 100%, the recall is 4.6%. We believe BioEx system is applicable for real use because a high level of precision is the key to BioEx success.

Acknowledgement: The authors thank Dr. Weiqing Wang for her contribution to this work. The authors also thank Michael Bales, Li Zhou and Eric Silfen, and three anonymous reviewers for valuable comments. The authors acknowledge the support of Juvenile Diabetes Foundation International (JDRF 6-2005-835).

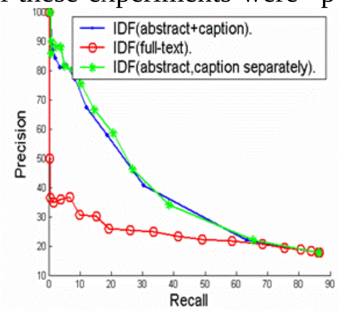


Figure 2

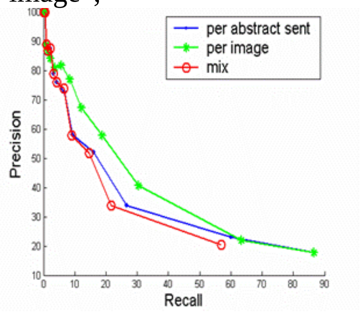


Figure 3

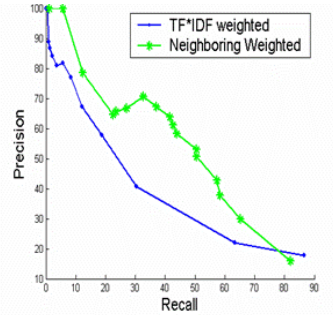


Figure 4

References:

Corpet F (1988) Multiple sequence alignment with hierarchical clustering. Nucleic Acids Res 16:10881-10890

Hearst M (2003) The BioText project. A powerpoint presentation.

Herrero J, Valencia A, Dopazo J (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics 17:126-136

Lee M, Wang W, Yu H (2006) Exploring supervised and unsupervised methods to detect topics in Biomedical text. BMC Bioinformatics 7:140

Smalheiser NR, Swanson DR (1998) Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. Comput Methods Programs Biomed 57:149-153

Subword-based Tagging by Conditional Random Fields for Chinese Word Segmentation

Ruiqiang Zhang[1,2] and **Genichiro Kikui**[*] and **Eiichiro Sumita**[1,2]
[1]National Institute of Information and Communications Technology
[2]ATR Spoken Language Communication Research Laboratories
2-2-2 Hikaridai, Seiika-cho, Soraku-gun, Kyoto, 619-0288, Japan
{ruiqiang.zhang,eiichiro.sumita}@atr.jp

Abstract

We proposed two approaches to improve Chinese word segmentation: a subword-based tagging and a confidence measure approach. We found the former achieved better performance than the existing character-based tagging, and the latter improved segmentation further by combining the former with a dictionary-based segmentation. In addition, the latter can be used to balance out-of-vocabulary rates and in-vocabulary rates. By these techniques we achieved higher F-scores in CITYU, PKU and MSR corpora than the best results from Sighan Bakeoff 2005.

1 Introduction

The character-based "IOB" tagging approach has been widely used in Chinese word segmentation recently (Xue and Shen, 2003; Peng and McCallum, 2004; Tseng et al., 2005). Under the scheme, each character of a word is labeled as 'B' if it is the first character of a multiple-character word, or 'O' if the character functions as an independent word, or 'I' otherwise." For example, "全(whole) 北京市(Beijing city)" is labeled as "全(whole)/O 北(north)/B 京(capital)/I 市(city)/I".

We found that so far all the existing implementations were using character-based IOB tagging. In this work we propose a subword-based IOB tagging, which assigns tags to a pre-defined lexicon subset consisting of the most frequent multiple-character words in addition to single Chinese characters. If only Chinese characters are used, the subword-based IOB tagging is downgraded into a character-based one. Taking the same example mentioned above, "全(whole) 北京市(Beijing city)" is labeled as "全(whole)/O 北京(Beijing)/B 市(city)/I" in the subword-based tagging, where "北京(Beijing)/B" is labeled as one unit. We will give a detailed description of this approach in Section 2.

In addition, we found a clear weakness with the IOB tagging approach: It yields a very low in-vocabulary (IV) rate (R-iv) in return for a higher out-of-vocabulary (OOV) rate (R-oov). In the results of the closed test in Bakeoff 2005 (Emerson, 2005), the work of (Tseng et al., 2005), using conditional random fields (CRF) for the IOB tagging, yielded very high R-oovs in all of the four corpora used, but the R-iv rates were lower. While OOV recognition is very important in word segmentation, a higher IV rate is also desired. In this work we propose a confidence measure approach to lessen the weakness. By this approach we can change R-oovs and R-ivs and find an optimal tradeoff. This approach will be described in Section 2.2.

In the followings, we illustrate our word segmentation process in Section 2, where the subword-based tagging is implemented by the CRFs method. Section 3 presents our experimental results. Section 4 describes current state-of-the-art methods for Chinese word segmentation, with which our results were compared. Section 5 provides the concluding remarks.

2 Our Chinese word segmentation process

Our word segmentation process is illustrated in Fig. 1. It is composed of three parts: a dictionary-based N-gram word segmentation for segmenting IV words, a subword-based tagging by the CRF for recognizing OOVs, and a confidence-dependent word segmentation used for merging the results of both the dictionary-based and the IOB tagging. An example exhibiting each step's results is also given in the figure.

Since the dictionary-based approach is a well-known method, we skip its technical descriptions. However, keep in mind that the dictionary-based approach can produce a higher R-iv rate. We will use this advantage in the confidence measure approach.

2.1 Subword-based IOB tagging using CRFs

There are several steps to train a subword-based IOB tagger. First, we extracted a word list from the training data sorted in decreasing order by their counts in the training

[*] Now the second author is affiliated with NTT.

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 193–196,
New York, June 2006. ©2006 Association for Computational Linguistics

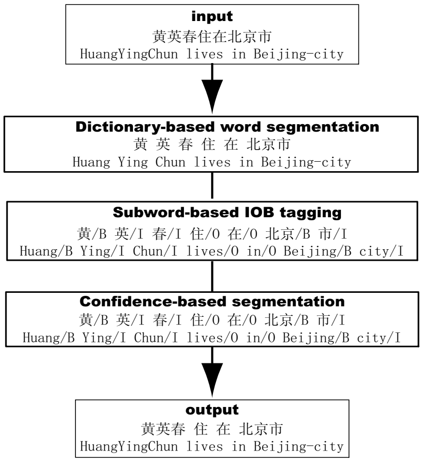

input
黄英春住在北京市
HuangYingChun lives in Beijing-city
Dictionary-based word segmentation
黄 英 春 住 在 北京市
Huang Ying Chun lives in Beijing-city
Subword-based IOB tagging
黄/B 英/I 春/I 住/0 在/0 北京/B 市/I
Huang/B Ying/I Chun/I lives/0 in/0 Beijing/B city/I
Confidence-based segmentation
黄/B 英/I 春/I 住/0 在/0 北京/B 市/I
Huang/B Ying/I Chun/I lives/0 in/0 Beijing/B city/I
output
黄英春 住 在 北京市
HuangYingChun lives in Beijing-city

Figure 1: Outline of word segmentation process

data. We chose all the single characters and the top multi-character words as a lexicon subset for the IOB tagging. If the subset consists of Chinese characters only, it is a character-based IOB tagger. We regard the words in the subset as the subwords for the IOB tagging.

Second, we re-segmented the words in the training data into subwords belonging to the subset, and assigned IOB tags to them. For a character-based IOB tagger, there is only one possibility of re-segmentation. However, there are multiple choices for a subword-based IOB tagger. For example, “北京市(Beijing-city)” can be segmented as “北京市(Beijing-city)/O,” or “北京(Beijing)/B 市(city)/I,” or ”北(north)/B 京(capital)/I 市(city)/I.” In this work we used forward maximal match (FMM) for disambiguation. Of course, backward maximal match (BMM) or other approaches are also applicable. We did not conduct comparative experiments because trivial differences of these approaches may not result in significant consequences to the subword-based approach.

In the third step, we used the CRFs approach to train the IOB tagger (Lafferty et al., 2001) on the training data. We downloaded and used the package “CRF++” from the site “http://www.chasen.org/t̃aku/software.” According to the CRFs, the probability of an IOB tag sequence, $T = t_0 t_1 \cdots t_M$, given the word sequence, $W = w_0 w_1 \cdots w_M$, is defined by

$$\begin{aligned} & p(T|W) = \\ & \exp\left(\sum_{i=1}^{M}\left(\sum_{k} \lambda_k f_k(t_{i-1}, t_i, W) + \sum_{k} \mu_k g_k(t_i, W)\right)\right) / Z, \quad (1) \\ & Z = \sum_{T=t_0 t_1 \cdots t_M} p(T|W) \end{aligned}$$

where we call $f_k(t_{i-1}, t_i, W)$ bigram feature functions because the features trigger the previous observation t_{i-1} and current observation t_i simultaneously; $g_k(t_i, W)$, the unigram feature functions because they trigger only current observation t_i. λ_k and μ_k are the model parameters corresponding to feature functions f_k and g_k respectively.

The model parameters were trained by maximizing the log-likelihood of the training data using L-BFGS gradient descent optimization method. In order to overcome overfitting, a gaussian prior was imposed in the training.

The types of unigram features used in our experiments included the following types:

$$w_0, w_{-1}, w_1, w_{-2}, w_2, w_0 w_{-1}, w_0 w_1, w_{-1} w_1, w_{-2} w_{-1}, w_2 w_0$$

where w stands for word. The subscripts are position indicators. 0 means the current word; $-1, -2$, the first or second word to the left; $1, 2$, the first or second word to the right.

For the bigram features, we only used the previous and the current observations, $t_{-1} t_0$.

As to feature selection, we simply used absolute counts for each feature in the training data. We defined a cutoff value for each feature type and selected the features with occurrence counts over the cutoff.

A forward-backward algorithm was used in the training and viterbi algorithm was used in the decoding.

2.2 Confidence-dependent word segmentation

Before moving to this step in Figure 1, we produced two segmentation results: the one by the dictionary-based approach and the one by the IOB tagging. However, neither was perfect. The dictionary-based segmentation produced results with higher R-ivs but lower R-oovs while the IOB tagging yielded the contrary results. In this section we introduce a confidence measure approach to combine the two results. We define a confidence measure, $CM(t_{iob}|w)$, to measure the confidence of the results produced by the IOB tagging by using the results from the dictionary-based segmentation. The confidence measure comes from two sources: IOB tagging and dictionary-based word segmentation. Its calculation is defined as:

$$CM(t_{iob}|w) = \alpha CM_{iob}(t_{iob}|w) + (1 - \alpha)\delta(t_w, t_{iob})_{ng} \quad (2)$$

where t_{iob} is the word w's IOB tag assigned by the IOB tagging; t_w, a prior IOB tag determined by the results of the dictionary-based segmentation. After the dictionary-based word segmentation, the words are re-segmented into subwords by FMM before being fed to IOB tagging. Each subword is given a prior IOB tag, t_w. $CM_{iob}(t|w)$, a confidence probability derived in the process of IOB tagging, is defined as

$$CM_{iob}(t|w_i) = \frac{\sum_{T=t_0 t_1 \cdots t_M, t_i = t} P(T|W, w_i)}{\sum_{T=t_0 t_1 \cdots t_M} P(T|W)}$$

where the numerator is a sum of all the observation sequences with word w_i labeled as t.

$\delta(t_w, t_{iob})_{ng}$ denotes the contribution of the dictionary-based segmentation. It is a Kronecker delta function defined as

$$\delta(t_w, t_{iob})_{ng} = \{ \begin{array}{ll} 1 & \texttt{if } t_w = t_{iob} \\ 0 & \texttt{otherwise} \end{array}$$

In Eq. 2, α is a weighting between the IOB tagging and the dictionary-based word segmentation. We found the value 0.7 for α, empirically.

By Eq. 2 the results of IOB tagging were re-evaluated. A confidence measure threshold, t, was defined for making a decision based on the value. If the value was lower than t, the IOB tag was rejected and the dictionary-based segmentation was used; otherwise, the IOB tagging segmentation was used. A new OOV was thus created. For the two extreme cases, $t = 0$ is the case of the IOB tagging while $t = 1$ is that of the dictionary-based approach. In a real application, a satisfactory tradeoff between R-ivs and R-oovs could find through tuning the confidence threshold. In Section 3.2 we will present the experimental segmentation results of the confidence measure approach.

3 Experiments

We used the data provided by Sighan Bakeoff 2005 to test our approaches described in the previous sections. The data contain four corpora from different sources: Academia Sinica (AS), City University of Hong Kong (CITYU), Peking University (PKU) and Microsoft Research in Beijing (MSR). Since this work was to evaluate the proposed subword-based IOB tagging, we carried out the closed test only. Five metrics were used to evaluate segmentation results: recall(R), precision(P), F-score(F), OOV rate(R-oov) and IV rate(R-iv). For detailed info. of the corpora and these scores, refer to (Emerson, 2005).

For the dictionary-based approach, we extracted a word list from the training data as the vocabulary. Trigram LMs were generated using the SRI LM toolkit for disambiguation. Table 1 shows the performance of the dictionary-based segmentation. Since there were some single-character words present in the test data but not in the training data, the R-oov rates were not zero in this experiment. In fact, there were no OOV recognition. Hence, this approach produced lower F-scores. However, the R-ivs were very high.

3.1 Effects of the Character-based and the subword-based tagger

The main difference between the character-based and the word-based is the contents of the lexicon subset used for re-segmentation. For the character-based tagging, we used all the Chinese characters. For the subword-based tagging, we added another 2000 most frequent multiple-character words to the lexicons for tagging. The segmentation results of the dictionary-based were re-segmented using the FMM, and then labeled with "IOB" tags by the CRFs. The segmentation results using CRF tagging are shown in Table 2, where the upper numbers of each slot were produced by the character-based approach while the lower numbers were of the subword-based. We found that the proposed subword-based approaches were effective in CITYU and MSR corpora, raising the F-scores from 0.941 to 0.946 for CITYU corpus, 0.959 to 0.964 for MSR corpus. There were no F-score changes for AS and PKU corpora, but the recall rates were improved. Comparing Table 1 and 2, we found the CRF-modeled IOB tagging yielded better segmentation than the dictionary-based approach. However, the R-iv rates were getting worse in return for higher R-oov rates. We will tackle this problem by the confidence measure approach.

	R	P	F	R-oov	R-iv
AS	0.941	0.881	0.910	0.038	0.982
CITYU	0.928	0.851	0.888	0.164	0.989
PKU	0.948	0.912	0.930	0.408	0.981
MSR	0.968	0.927	0.947	0.048	0.993

Table 1: Our segmentation results by the dictionary-based approach for the closed test of Bakeoff 2005, very low R-oov rates due to no OOV recognition applied.

	R	P	F	R-oov	R-iv
AS	0.951 0.953	0.942 0.940	0.947 0.947	0.678 0.647	0.964 0.967
CITYU	0.939 0.950	0.943 0.942	0.941 0.946	0.700 0.736	0.958 0.967
PKU	0.940 0.943	0.950 0.946	0.945 0.945	0.783 0.754	0.949 0.955
MSR	0.957 0.965	0.960 0.963	0.959 0.964	0.710 0.716	0.964 0.972

Table 2: Segmentation results by a pure subword-based IOB tagging. The upper numbers are of the character-based and the lower ones, the subword-based.

3.2 Effect of the confidence measure

In section 2.2, we proposed a confidence measure approach to re-evaluate the results of IOB tagging by combinations of the results of the dictionary-based segmentation. The effect of the confidence measure is shown in Table 3, where we used $\alpha = 0.7$ and confidence threshold $t = 0.8$. In each slot, the numbers on the top were of the character-based approach while the numbers on the bottom were the subword-based. We found the results in Table 3 were better than those in Table 2 and Table 1, which prove that using confidence measure approach achieved the best performance over the dictionary-based segmentation and the IOB tagging approach. The act of confidence measure made a tradeoff between R-ivs and R-oovs, yielding higher R-oovs than Table 1 and higher R-

	R	P	F	R-oov	R-iv
AS	0.953	0.944	0.948	0.607	0.969
	0.956	0.947	0.951	0.649	0.969
CITYU	0.943	0.948	0.946	0.682	0.964
	0.952	0.949	0.951	0.741	0.969
PKU	0.942	0.957	0.949	0.775	0.952
	0.947	0.955	0.951	0.748	0.959
MSR	0.960	0.966	0.963	0.674	0.967
	0.972	0.969	0.971	0.712	0.976

Table 3: Effects of combination using the confidence measure. The upper numbers and the lower numbers are of the character-based and the subword-based, respectively

	AS	CITYU	MSR	PKU
Bakeoff-best	0.952	0.943	0.964	0.950
Ours	0.951	0.951	0.971	0.951

Table 4: Comparison our results with the best ones from Sighan Bakeoff 2005 in terms of F-score

ivs than Table 2.

Even with the use of confidence measure, the word-based IOB tagging still outperformed the character-based IOB tagging. It proves the proposed word-based IOB tagging was very effective.

4 Discussion and Related works

The IOB tagging approach adopted in this work is not a new idea. It was first used in Chinese word segmentation by (Xue and Shen, 2003), where maximum entropy methods were used. Later, this approach was implemented by the CRF-based method (Peng and McCallum, 2004), which was proved to achieve better results than the maximum entropy approach because it can solve the label bias problem (Lafferty et al., 2001).

Our main contribution is to extend the IOB tagging approach from being a character-based to a subword-based. We proved the new approach enhanced the word segmentation significantly. Our results are listed together with the best results from Bakeoff 2005 in Table 4 in terms of F-scores. We achieved the highest F-scores in CITYU, PKU and MSR corpora. We think our proposed subword-based tagging played an important role for the good results. Since it was a closed test, some information such as Arabic and Chinese number and alphabetical letters cannot be used. We could yield a better results than those shown in Table 4 using such information. For example, inconsistent errors of foreign names can be fixed if alphabetical characters are known. For AS corpus, “Adam Smith” are two words in the training but become a one-word in the test, “AdamSmith”. Our approaches produced wrong segmentations for labeling inconsistency.

Another advantage of the word-based IOB tagging over the character-based is its speed. The subword-based approach is faster because fewer words than characters were labeled. We found a speed up both in training and test.

The idea of using the confidence measure has appeared in (Peng and McCallum, 2004), where it was used to recognize the OOVs. In this work we used it more delicately. By way of the confidence measure we combined results from the dictionary-based and the IOB-tagging-based and as a result, we could achieve the optimal performance.

5 Conclusions

In this work, we proposed a subword-based IOB tagging method for Chinese word segmentation. Using the CRFs approaches, we prove that it outperformed the character-based method using the CRF approaches. We also successfully employed the confidence measure to make a confidence-dependent word segmentation. This approach is effective for performing desired segmentation based on users’ requirements to R-oov and R-iv.

Acknowledgements

The authors appreciate the reviewers’ effort and good advice for improving the paper.

References

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, Jeju, Korea.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML-2001*, pages 591–598.

Fuchun Peng and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proc. of Coling-2004*, pages 562–568, Geneva, Switzerland.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for Sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, Jeju, Korea.

Nianwen Xue and Libin Shen. 2003. Chinese word segmentation as LMR tagging. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*.

Comparing the roles of textual, acoustic and spoken-language features on spontaneous-conversation summarization

Xiaodan Zhu **Gerald Penn**

Department of Computer Science, University of Toronto
10 Kings College Rd., Toronto, Canada
{xzhu, gpenn} @cs.toronto.edu

Abstract

This paper is concerned with the summarization of spontaneous conversations. Compared with broadcast news, which has received intensive study, spontaneous conversations have been less addressed in the literature. Previous work has focused on textual features extracted from transcripts. This paper explores and compares the effectiveness of both textual features and speech-related features. The experiments show that these features incrementally improve summarization performance. We also find that speech disfluencies, which have been removed as noise in previous work, help identify important utterances, while the structural feature is less effective than it is in broadcast news.

1 Introduction

Spontaneous conversations are a very important type of speech data. Distilling important information from them has commercial and other importance. Compared with broadcast news, which has received the most intensive studies (Hori and Furui, 2003; Christensen et al. 2004; Maskey and Hirschberg, 2005), spontaneous conversations have been less addressed in the literature.

Spontaneous conversations are different from broadcast news in several aspects: (1) spontaneous conversations are often less well formed linguistically, e.g., containing more speech disfluencies and false starts; (2) the distribution of important utterances in spontaneous conversations could be different from that in broadcast news, e.g., the beginning part of news often contains important information, but in conversations, information may be more evenly distributed; (3) conversations often contain discourse clues, e.g., question-answer pairs and speakers' information, which can be utilized to keep the summary coherent; (4) word error rates (WERs) from speech recognition are usually much higher in spontaneous conversations.

Previous work on spontaneous-conversation summarization has mainly focused on textual features (Zechner, 2001; Gurevych and Strube, 2004), while speech-related features have not been explored for this type of speech source. This paper explores and compares the effectiveness of both textual features and speech-related features. The experiments show that these features incrementally improve summarization performance. We also discuss problems (1) and (2) mentioned above. For (1), Zechner (2001) proposes to detect and remove false starts and speech disfluencies from transcripts, in order to make the text-format summary concise and more readable. Nevertheless, it is not always necessary to remove them. One reason is that original utterances are often more desired to ensure comprehensibility and naturalness if the summaries are to be delivered as excerpts of audio (see section 2), in order to avoid the impact of WER. Second, disfluencies are not necessarily noise; instead, they show regularities in a number of dimensions (Shriberg, 1994), and correlate with many factors including topic difficulty (Bortfeld et al, 2001). Rather than removing them, we explore the effects of disfluencies on summarization, which, to our knowledge, has not yet been addressed in the literature. Our experiments show that they improve summarization performance.

To discuss problem (2), we explore and compare both textual features and speech-related features, as they are explored in broadcast news (Maskey and Hirschberg, 2005). The experiments show that the structural feature (e.g. utterance position) is less effective for summarizing spontaneous conversations than it is in broadcast news. MMR

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 197–200,
New York, June 2006. ©2006 Association for Computational Linguistics

and lexical features are the best. Speech-related features follow. The structural feature is least effective. We do not discuss problem (3) and (4) in this paper. For problem (3), a similar idea has been proposed to summarize online blogs and discussions. Problem (4) has been partially addressed by (Zechner & Waibel, 2000); but it has not been studied together with acoustic features.

2 Utterance-extraction-based summarization

Still at its early stage, current research on speech summarization targets a less ambitious goal: conducting extractive, single-document, generic, and surface-level-feature-based summarization. The pieces to be extracted could correspond to words (Koumpis, 2002; Hori and Furui, 2003). The extracts could be utterances, too. Utterance selection is useful. First, it could be a preliminary stage applied before word extraction, as proposed by Kikuchi et al. (2003) in their two-stage summarizer. Second, with utterance-level extracts, one can play the corresponding audio to users, as with the speech-to-speech summarizer discussed in Furui et al. (2003). The advantage of outputting audio segments rather than transcripts is that it avoids the impact of WERs caused by automatic speech recognition (ASR). We will focus on utterance-level extraction, which at present appears to be the only way to ensure comprehensibility and naturalness if the summaries are to be delivered as excerpts of audio themselves.

Previous work on spontaneous conversations mainly focuses on using textual features. Gurevych & Strube (2004) develop a shallow knowledge-based approach. The noun portion of WordNet is used as a knowledge source. The noun senses were manually disambiguated rather than automatically. Zechner (2001) applies maximum marginal relevance (MMR) to select utterances for spontaneous conversation transcripts.

3 Classification based utterance extraction

Spontaneous conversations contain more information than textual features. To utilize these features, we reformulate the utterance selection task as a binary classification problem, an utterance is either labeled as "1" (in-summary) or "0" (not-in-summary). Two state-of-the-art classifiers, support vector machine (SVM) and logistic regression (LR), are used. SVM seeks an optimal separating hyperplane, where the margin is maximal. In our experiments, we use the OSU-SVM package. Logistic regression (LR) is indeed a softmax linear regression, which models the posterior probabilities of the class label with the softmax of linear functions of feature vectors. For the binary classification that we require in our experiments, the model format is simple.

3.1 Features

The features explored in this paper include:

(1) MMR score: the score calculated with MMR (Zechner, 2001) for each utterance.

(2) Lexicon features: number of named entities, and utterance length (number of words). The number of named entities includes: person-name number, location-name number, organization-name number, and the total number. Named entities are annotated automatically with a dictionary.

(3) Structural features: a value is assigned to indicate whether a given utterance is in the first, middle, or last one-third of the conversation. Another Boolean value is assigned to indicate whether this utterance is adjacent to a speaker turn or not.

(4) Prosodic features: we use basic prosody: the maximum, minimum, average and range of energy, as well as those of fundamental frequency, normalized by speakers. All these features are automatically extracted.

(5) Spoken-language features: the spoken-language features include number of repetitions, filled pauses, and the total number of them. Disfluencies adjacent to a speaker turn are not counted, because they are normally used to coordinate interaction among speakers. Repetitions and pauses are detected in the same way as described in Zechner (2001).

4 Experimental results

4.1 Experiment settings

The data used for our experiments come from SWITCHBOARD. We randomly select 27 conversations, containing around 3660 utterances. The important utterances of each conversation are manually annotated. We use f-score and the ROUGE score as evaluation metrics. Ten-fold cross validation is applied to obtain the results presented in this section.

4.2 Summarization performance

4.2.1 F-score

Table-1 shows the f-score of logistic regression (LR) based summarizers, under different compression ratios, and with incremental features used.

	10%	15%	20%	25%	30%
(1) MMR	.246	.309	.346	.355	.368
(2) (1) +lexicon	.293	.338	.373	.380	.394
(3) (2)+structure	.334	.366	.400	.409	.404
(4) (3)+acoustic	**.336**	.364	.388	.410	.415
(5) (4)+spoken language	.333	**.376**	**.410**	**.431**	**.422**

Table 1. f-score of LR summarizers using incremental features

Below is the f-score of SVM-based summarizer:

	10%	15%	20%	25%	30%
(1) MMR	.246	.309	.346	.355	.368
(2) (1) +lexicon	.281	.338	.354	.358	.377
(3) (2)+structural	.326	.371	.401	.409	.408
(4) (3)+acoustic	.337	**.380**	.400	.422	.418
(5) (4)+spoken language	**.353**	**.380**	**.416**	**.424**	**.423**

Table 2. f-score of SVM summarizers using incremental features

Both tables show that the performance of summarizers improved, in general, with more features used. The use of lexicon and structural features outperforms MMR, and the speech-related features, acoustic features and spoken language features produce additional improvements.

4.2.2 ROUGE

The following tables provide the ROUGE-1 scores:

	10%	15%	20%	25%	30%
(1) MMR	.585	.563	.523	.492	.467
(2) (1) +lexicon	.602	.579	.543	.506	.476
(3) (2)+structure	**.621**	.591	.553	.516	.482
(4) (3)+acoustic	.619	.594	.554	.519	.485
(5) (4)+spoken language	.619	**.600**	**.566**	**.530**	**.492**

Table 3. ROUGE-1 of LR summarizers using incremental features

	10%	15%	20%	25%	30%
(1) MMR	.585	.563	.523	.492	.467
(2) (1) +lexicon	.604	.581	.542	.504	.577
(3) (2)+structure	.617	.600	.563	.523	.490
(4) (3)+acoustic	**.629**	.610	.573	.533	.496
(5)(4)+spoken language	.628	**.611**	**.576**	**.535**	**.502**

Table 4. ROUGE-1 of SVM summarizers using incremental features

The ROUGE-1 scores show similar tendencies to the f-scores: the rich features improve summarization performance over the baseline MMR summarizers. Other ROUGE scores like ROUGE-L show the same tendency, but are not presented here due to the space limit.

Both the f-score and ROUGE indicate that, in general, rich features incrementally improve summarization performance.

4.3 Comparison of features

To study the effectiveness of individual features, the receiver operating characteristic (ROC) curves of these features are presented in Figure-1 below. The larger the area under a curve is, the better the performance of this feature is. To be more exact, the definition for the y-coordinate (sensitivity) and the x-coordinate (1-specificity) is:

$$sensitivity = \frac{TP}{TP + FN} = true\ positive\ rate$$

$$specificity = \frac{TN}{TN + FP} = true\ negtive\ rate$$

where TP, FN, TN and FP are true positive, false negative, true negative, and false positive, respectively.

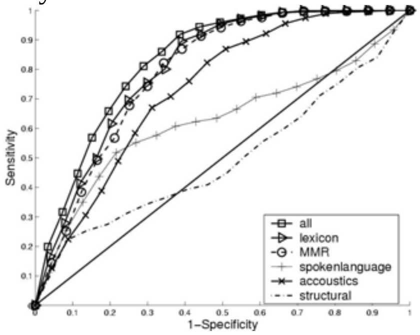


Figure-1. ROC curves for individual features

Lexicon and MMR features are the best two individual features, followed by spoken-language and acoustic features. The structural feature is least effective.

Let us first revisit the problem (2) discussed above in the introduction. The effectiveness of the structural feature is less significant than it is in broadcast news. According to the ROC curves presented in Christensen et al. (2004), the structural feature (utterance position) is one of the best features for summarizing read news stories, and is less effective when news stories contain spontaneous speech. Both their ROC curves cover larger area than the structural feature here in figure 1, that is, the structure feature is less effective for summarizing spontaneous conversation than it is in broadcast news. This reflects, to some extent, that information is more evenly distributed in spontaneous conversations.

Now let us turn to the role of speech disfluencies, which are very common in spontaneous conversations. Previous work detects and removes disfluencies as noise. Indeed, disfluencies show regularities in a number of dimensions (Shriberg, 1994). They correlate with many factors including the topic difficulty (Bortfeld et al, 2001). Tables 1-4 above show that they improve summarization performance when added upon other features. Figure-1 shows that when used individually, they are better than the structural feature, and also better than acoustic features at the left 1/3 part of the figure, where the summary contains relatively fewer utterances. Disfluencies, e.g., pauses, are often inserted when speakers have word-searching problem, e.g., a problem finding topic-specific keywords:

Speaker A: with all the ***uh*** ***sulfur*** *and all that other stuff they're dumping out into the atmosphere.*

The above example is taken from a conversation that discusses pollution. The speaker inserts a filled pause *uh* in front of the word *sulfur*. Pauses are not randomly inserted. To show this, we remove them from transcripts. Section-2 of SWITCHBOARD (about 870 dialogues and 189,000 utterances) is used for this experiment. Then we insert these pauses back randomly, or insert them back at their original places, and compare the difference. For both cases, we consider a window with 4 words after each filled pause. We average the tf.idf scores of the words in each of these windows. Then, for all speaker-inserted pauses, we obtain a set of averaged tf.idf scores. And for all randomly-inserted pauses, we have another set. The mean of the former set (5.79 in table 5) is statistically higher than that of the latter set (5.70 in table 5). We can adjust the window size to 3, 2 and 1, and then get the following table.

Window size		1	2	3	4
Mean of tf.idf score	Insert Randomly	5.69	5.69	5.70	5.70
	Insert by speaker	5.72	5.82	5.81	5.79
Difference is significant? (t-test, $p<0.05$)		Yes	Yes	Yes	Yes

Table 5. Average tf.idf scores of words following filled pauses.

The above table shows that instead of randomly inserting pauses, real speakers insert them in front of words with higher tf.idf scores. This helps explain why disfluencies work.

5 Conclusions

Previous work on summarizing spontaneous conversations has mainly focused on textual features. This paper explores and compares both textual and speech-related features. The experiments show that these features incrementally improve summarization performance. We also find that speech disfluencies, which are removed as noise in previous work, help identify important utterances, while the structural feature is less effective than it is in broadcast news.

6 References

Bortfeld, H., Leon, S.D., Bloom, J.E., Schober, M.F., & Brennan, S.E. 2001. Disfluency Rates in Conversation: Effects of Age, Relationship, Topic Role, and Gender. Language and Speech, 44(2): 123-147

Christensen, H., Kolluru, B., Gotoh, Y., Renals, S., 2004. From text summarisation to style-specific summarisation for broadcast news. Proc. ECIR-2004.

Furui, S., Kikuichi T. Shinnaka Y., and Hori C. 2003. Speech-to-speech and speech to text summarization,. First International workshop on Language Understanding and Agents for Real World Interaction, 2003.

Gurevych I. and Strube M. 2004. Semantic Similarity Applied to Spoken Dialogue Summarization. COLING-2004.

Hori C. and Furui S., 2003. A New Approach to Automatic Speech Summarization IEEE Transactions on Multimedia, Vol. 5, NO. 3, September 2003,

Kikuchi T., Furui S. and Hori C., 2003. Automatic Speech Summarization Based on Sentence Extraction and Compaction, Proc. ICASSP-2003.

Koumpis K., 2002. Automatic Voicemail Summarisation for Mobile Messaging Ph.D. Thesis, University of Sheffield, UK, 2002.

Maskey, S.R., Hirschberg, J. "Comparing Lexial, Acoustic/Prosodic, Discourse and Structural Features for Speech Summarization", Eurospeech 2005.

Shriberg, E.E. (1994). Preliminaries to a Theory of Speech Disfluencies. Ph.D. thesis, University of California at Berkeley.

Zechner K. and Waibel A., 2000. Minimizing word error rate in textual summaries of spoken language. NAACL-2000.

Zechner K., 2001. Automatic Summarization of Spoken Dialogues in Unrestricted Domains. Ph.D. thesis, Carnegie Mellon University, November 2001.

Bridging the Inflection Morphology Gap for Arabic Statistical Machine Translation

Andreas Zollmann and **Ashish Venugopal** and **Stephan Vogel**
School of Computer Science
Carnegie Mellon University
{zollmann,ashishv,stephan.vogel}@cs.cmu.edu

Abstract

Statistical machine translation (SMT) is based on the ability to effectively learn word and phrase relationships from parallel corpora, a process which is considerably more difficult when the extent of morphological expression differs significantly across the source and target languages. We present techniques that select appropriate word segmentations in the morphologically rich source language based on contextual relationships in the target language. Our results take advantage of existing word level morphological analysis components to improve translation quality above state-of-the-art on a limited-data Arabic to English speech translation task.

1 Introduction

The problem of translating from a language exhibiting rich inflectional morphology to a language exhibiting relatively poor inflectional morphology presents several challenges to the existing components of the statistical machine translation (SMT) process. This inflection gap causes an abundance of surface word forms [1] in the source language compared with relatively few forms in the target language. This mismatch aggravates several issues found in natural language processing: more unknown words forms in unseen data, more words occurring only once, more distinct words and lower token-to-type ratios (mean number of occurrences over all distinct words) in the source language than in the target language.

Lexical relationships under the standard IBM models (Brown et al., 1993) do not account for many-to-many mappings, and phrase extraction relies heavily on the accuracy of the IBM word-to-word alignment. In this work, we propose an approach to bridge the inflectional gap that addresses the issues described above through a series of preprocessing steps based on the Buckwalter Arabic Morphological Analyzer (BAMA) tool (Buckwalter, 2004). While (Lee et al., 2003) develop accurate segmentation models of Arabic surface word forms using manually segmented data, we rely instead on the translated context in the target language, leveraging the manually constructed lexical gloss from BAMA to select the appropriate segmented sense for each Arabic source word.

Our technique, applied as preprocessing to the source corpus, splits and normalizes surface words based on the target sentence context. In contrast to (Popovic and Ney, 2004) and (Nießen and Ney, 2004), we do not modify the IBM models, and we leave reordering effects to the decoder. Statistically significant improvements (Zhang and Vogel, 2004) in BLEU and NIST translation score over a lightly stemmed baseline are reported on the available and well known BTEC IWSLT'05 Arabic-English corpus (Eck and Hori, 2005).

[1] We use the term surface form to refer to a series of characters separated by whitespace

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 201–204,
New York, June 2006.

2 Arabic Morphology in Recent Work

Arabic-to-English machine translation exemplifies some of the issues caused by the inflection gap. Refer to (Buckwalter, 2005) and (Larkey et al., 2002) for examples that highlight morphological inflection for a simple Modern Standard Arabic (MSA) word and basic stemming operations that we use as our baseline system.

(Nießen and Ney, 2000) tackle the inflection gap for German-to-English word alignment by performing a series of morphological operations on the German text. They fragment words based on a full morphological analysis of the sentence, but need to use domain specific and hand written rules to deal with ambiguous fragmentation. (Nießen and Ney, 2004) also extend the corpus by annotating each source word with morphological information and building a hierarchical lexicon. The experimental results show dramatic improvements from sentence-level restructuring (question inversion, separated verb prefixes and merging phrases), but limited improvement from the hierarchical lexicon, especially as the size of the training data increases.

We conduct our morphological analysis at the word level, using Buckwalter Arabic Morphological Analyzer (BAMA) version 2.0 (Buckwalter, 2004). BAMA analyzes a given surface word, returning a set of potential segmentations (order of a dozen) for the source word into prefixes, stems, and suffixes. Our techniques select the appropriate splitting from that set by taking into account the target sides (full sentences) of that word's occurrences in the training corpus. We now describe each splitting technique that we apply.

2.1 BAMA: Simple fragment splitting

We begin by simply replacing each Arabic word with the fragments representing the first of the possible splittings returned by the BAMA tool. BAMA uses simple word-based heuristics to rank the splitting alternatives.

2.2 CONTEXT: Single Sense selection

In the step CONTEXT, we take advantage of the gloss information provided in BAMA's lexicon. Each potential splitting corresponds to a particular choice of prefix, stem and suffix, all of which exist in the manually constructed lexicon, along with a set of possible translations (*glosses*) for each fragment. We select a fragmentation (choice of splitting for the source word) whose corresponding glosses have the most target side matches in the parallel translation (of the full sentence). The choice of fragmentation is saved and used for all occurrences of the surface form word in training and testing, introducing context sensitivity without parsing solutions. In case of unseen words during testing, we segment it simply using the first alternative from the BAMA tool. This allows us to still translate an unseen test word correctly even if the surface form was never seen during training.

2.3 CORRMATCH: Correspondence matching

The Arabic language often encodes linguistic information within the surface word form that is not present in English. Word fragments that represent this missing information are misleading in the translation process unless explicitly aligned to the NULL word on the target side. In this step we explicitly remove fragments that correspond to lexical information that is not represented in English. While (Lee, 2004) builds part of speech models to recognize such elements, we use the fact that their corresponding English translations in the BAMA lexicon are empty. Examples of such fragments are case and gender markers. As an example of CORRMATCH removal, we present the Arabic sentence " h'*A lA ya zAl u gayor naZiyf " (after BAMA only) which becomes "h'*A lA ya zAl _ gayor naZiyf" after the CORRMATCH stage. The "u" has been removed.

3 Experimental Framework

We evaluate the impact of inflectional splitting on the BTEC (Takezawa et al., 2002) IWSLT05 Arabic language data track. The "Supplied" data track includes a 20K Arabic/English sentence pair training set, as well as a development ("DevSet") and test ("Test05") set of 500 Arabic sentences each and 16 reference translations per Arabic sentence. Details regarding the IWSLT evaluation criteria and data topic and collection methods are available in (Eck and Hori, 2005). We also evaluate on test and development data randomly sampled from the complete supplied dev and test data, due to considerations noted by (Josep M.Crego, 2005) regarding the similarity of the development and test data sets.

3.1 System description

Translation experiments were conducted using the (Vogel et al., 2003) system with reordering and future cost estimation. We trained translation parameters for 10 scores (language model, word and phrase count, and 6 translation model scores from (Vogel, 2005)) with Minimum Error Rate training on the development set. We optimized separately for both the NIST (Doddington, 2002) and the BLEU metrics (Papineni et al., 2002).

4 Translation Results

Table 1 and 2 shows the results of each stage of inflectional splitting on the BLEU and NIST metrics. Basic orthographic normalization serves as a baseline (merging all Alif, tar marbuta, ee forms to the base form). The test set NIST scores show steady improvements of up to 5 percent relative, as more sophisticated splitting techniques are used, ie BAMA+CONTEXT+CORRMATCH. These improvements are statistically significant over the baseline in both metrics as measured by the techniques in (Zhang and Vogel, 2004).

Our NIST results for all the final stages of inflectional splitting would place us above the top NIST scores from the ISWLT evaluation on the supplied test set.[2] On both DevSet/Test05 and the randomly split data, we see more dramatic improvements in the NIST scores than in BLEU. This might be due to the NIST metric's sensitivity to correctly translating certain high gain words in the test corpus. Inflectional splitting techniques that cause previously unknown surface form words to be translated correctly after splitting can significantly impact the overall score.

5 Conclusion and Future Work

This work shows the potential for significant improvements in machine translation quality by directly bridging the inflectional gap across language pairs. Our method takes advantage of source and target language context when conducting morphological analysis of each surface word form, while avoiding complex parsing engines or refinements to the alignment training process. Our results are presented on moderately sized corpora rather than the scarce resource domain that has been traditionally employed to highlight the impact of detailed morphological analysis.

By showing the impact of simple processing steps we encourage the creation of simple word and gloss level analysis tools for new languages and show that small investments in this direction (compared to high octane context sensitive parsing tools) can yield dramatic improvements, especially when rapid development of machine translation tools becomes increasingly relevant to the research community. While our work focused on processing the morphologically rich language and then translating ”down” into the morphologically poor language, we plan to use the analysis tools developed here to model the reverse translation process as well, the harder task of translating ”up” into a highly inflected space.

References

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

Tim Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. LDC Catalog No. LDC2004L02, Linguistic Data Consortium, www.ldc.upenn.edu/Catalog.

Tim Buckwalter. 2005. www.qamus.org/morphology.htm.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *In Proc. ARPA Workshop on Human Language Technology*.

Matthias Eck and Chiori Hori. 2005. Overview of the IWSLT 2005 evaluation campaign. In *Proceedings of International Workshop on Spoken Language Translation*, pages 11–17.

Jose B.Marino Josep M.Crego, Adria de Gispert. 2005. The talp ngram-based smt system for iwslt'05. In *Proceedings of International Workshop on Spoken Language Translation*, pages 191–198.

[2] The IWSLT evaluation did not allow systems to train separately for evaluation on BLEU or NIST, but results from the proceedings indicate that top performers in each metric optimized towards the respective metric.

Inflection system	NIST – Dev.	**NIST – Test**	BLEU – Dev.	**BLEU – Test**
No preprocessing	9.33	9.44	51.1	49.7
Orthographic normalization (baseline)	9.41	9.51	51.0	49.7
BAMA	9.90	9.76 (+2.5%)	52.0	50.2 (+1%)
BAMA+CONTEXT+CORRMATCH	9.91	**10.02** (+5.3%)	52.8	**52.0** (+4.7%)

Table 1: Translation results for each stage of inflectional splitting for the merged, sampled dev. and test data, highest scores in bold, relative improvements in brackets

Inflection system	NIST – Dev.	**NIST – Test**	BLEU – Dev.	**BLEU – Test**
No preprocessing	9.46	9.38	51.1	49.6
Orthographic normalization (baseline)	9.58	9.35	52.1	49.8
BAMA	10.10	9.60 (+2.7%)	53.8	48.8 (-2%)
BAMA+CONTEXT+CORRMATCH	10.08	**9.79** (+4.7%)	53.7	**50.6** (+1.6%)

Table 2: Translation results for each stage of inflectional splitting for the BTEC Supplied DevSet/Test05 data, highest scores in bold, relative improvements in brackets

Leah Larkey, Lisa Ballesteros, and Margaret Connell. 2002. Improving stemming for arabic information retrieval: Light stemming and co-occurrence analysis. In *Proc. of the 25th annual international ACM SIGIR conference on Research and development information retrieval*.

Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. 2003. Language model based arabic word segmentation. In *ACL*, Sapporo, Japan, July 6-7.

Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Boston,MA, May 27-June 1.

Sonja Nießen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *The 18th International Conference on Computational Linguistics*.

Sonja Nießen and Herman Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Comput. Linguist.*, 30(2):181–204.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association of Computational Linguistics*, pages 311–318.

H. Popovic and Hermann Ney. 2004. Improving word alignment quality using morpho-syntactic information. In *20th International Conference on Computational Linguistics (CoLing), Geneva, Switzerland.*

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proc. of LREC 2002*, pages 147–152, Las Palmas, Canary Islands, Spain, May.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, and Alex Waibel. 2003. The CMU statistical translation system. In *Proceedings of MT Summit IX*, New Orleans, LA, September.

Stephan Vogel. 2005. PESA: Phrase pair extraction as sentence splitting. In *Proceedings of MT Summit X*, Phuket,Thailand, September.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMII)*, Baltimore, MD, October.

Author Index