IJCLA **ISSN 0976-0962**

International Journal of Computational Linguistics and Applications

Vol. 2 No. 1-2 Jan-Dec 2011

Editor-in-Chief
Alexander Gelbukh

ISSN 0976-0962

International Journal of Computational Linguistics and Applications

Vol. 2 **No. 1-2** **Jan-Dec 2011**

International Journal of Computational Linguistics and Applications – IJCLA (started in 2010) is a peer-reviewed international journal published twice a year, in March and September. It publishes original research papers related to computational linguistics, natural language processing, human language technologies and their applications.

The views expressed herein are those of the authors. *International Journal of Computational Linguistics and Applications* reserves the right to edit the material.

Subscription: India: Rs. 1999
Rest of the world: US$ 150

Payments can be made by Cheques/Bank Drafts/International Money Orders drawn in the name of BAHRI PUBLICATIONS, NEW DELHI and sent to:

BAHRI PUBLICATIONS
1749A/5, 1st Floor, Gobindpuri Extension,
P. O. Box 4453, Kalkaji, New Delhi 110019
Telephones: 011-65810766, (0) 9811204673, (0) 9212794543
E-mail: bahrius@vsnl.com; bahripublications@yahoo.com
Website: http://www.bahripublications.com

Printed & Published by Deepinder Singh Bahri, for and on behalf of **BAHRI PUBLICATIONS**, New Delhi.

International Journal of Computational Linguistics and Applications

Vol. 2 **No. 1-2** **Jan-Dec 2011**

CONTENTS

Lexical Resources and Text Analysis

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 9–24
RECEIVED 01/11/10 ACCEPTED 21/12/10 FINAL 11/04/11

Large-vocabulary Lexical Choice with Rich Context Features

YUSUKE MATSUBARA[1] AND JUN'ICHI TSUJII[1,2]

[1] *The University of Tokyo, Japan*
[2] *Manchester Interdisciplinary Biocentre, UK*

ABSTRACT

This paper shows that syntactic information improves large-scale statistical lexical choice. Given a set of possible words to use, statistical lexical choice is the task to choose the most appropriate word to fill a gap in a sentence. The state-of-the-art methods of statistical lexical choice either rely only on window-based cooccurrence information or are focused on to specific word classes. We present a discriminative model of statistical lexical choice with local syntactic features and document-level features, in addition to window-based features. We evaluated our systems in the setting where we try to select the best substitution candidates for all occurrences of the content words, as well as in the smaller evaluation sets used in previous works. Experimental results on Penn Treebank and BLLIP corpus showed that the proposed method outperformed the state-of-the-art methods and that syntactic features improved the performance of prediction of lexical choice.

1 INTRODUCTION

Choosing a right word that conveys the meaning in mind is a difficult task, even for human. We encounter similar challenges in constructing systems in various application areas of natural language processing, including text information retrieval, machine translation and natural language generation. In most text collections including highly specialized ones, writers

may use different expressions to denote the same concept. This issue, which is known as synonymy or lexical variation, has significant importance in improving coverage of IR systems. In machine translation, the selection of translation words is an important subtask, especially when the output of the system should fit to a specific style or controlled vocabulary [1]. Another straightforward application is the lexical chooser in natural language generation systems [2]. A lexical chooser chooses the most appropriate lexical entry from the lexicon, given a semantic representation of the text to be generated. Having an accurate lexical chooser in natural language generation is important especially when the size of the vocabulary is large.

The task of choosing a right word given a context, which we call *lexical choice*, strongly relates to paraphrasing. In both of the two areas, we try to model synonymy of the words that occur in corpora. The results of paraphrasing methods are usually evaluated by comparing each set of the output expressions with those taken from thesauri constructed by human. It means that they aim to build context-independent knowledge of synonymy. While paraphrasing takes more importance on the aspect of unsupervised mining of synonymous expressions from corpora, lexical choice focuses on how to filter true synonyms from a set of substitution candidates, given a specific context.

To obtain accurate models of synonymy, it is necessary to capture context information. Strictly speaking, it is almost impossible to have a perfectly-interchangeable set of expressions in natural language. [3] For example, people may accept that the verb *command* can be replaced with *tell* in a sentence of military-related context, such as *The general* commanded/told *the officers that* .... The two words *command* and *tell* are obviously non-interchangeable in general context, but can be interchangeable if the context supports that substitution.

This motivates the statistical modeling of lexical choice, which aims, unlike traditional context-insensitive approaches to paraphrasing, to find which set of expression can be used interchangeably, given a specific context. A more formal description of lexical substitution will be given in Section 2.

Previous researches on lexical choice mostly focused on either the use of local context or limited coverage of vocabulary, as discussed in Section 3. In this paper, we propose to use wider context that spans over

[3] In this paper, we refer to the task defined in Section 2 as *lexical choice*, *lexical substitution* or *context-sensitive lexical parphrasing*, following different terminologies of previous researches.

syntactic phrases and a document, and evaluate with a more realistic size of vocabulary.

2 TASK DESCRIPTION

Following [3] and [4], we define the task of lexical choice as follows.

We are given a sentence, candidate substitutions, and a *lexical gap*, or a position which has to be filled with one of the candidate substitutions. Our task is to choose the most appropriate word from the candidate substitutions considering the context surrounding the lexical gap.

Let us illustrate the task description with an example of a computer-aided writing system composed of a paraphraser and lexical chooser. A user of the system inputs a sentence " *Workers dumped large burlap sacks of the imported substance*$_{[p_1]}$ *into a huge bin*." In the sentence, with a mark $[p_1]$ he tells the system that he has low confidence in his selection of the marked word and wants the system to suggest better alternatives. The system generates candidate expressions for the marked position, or *lexical gap* denoted by $[p_1]$. First, the paraphraser retrieves candidate substitutions, that is, expressions that are synonymous to the input expression *substance*, such as *substance*, *stuff* and *material*. Then, from the candidate substitutions, the lexical chooser chooses the substitution that is most probable to be placed in the lexical gap, typically by exploiting the information from available large scale corpora.

3 PREVIOUS WORK

In this section, we summarize existing approaches to context sensitive lexical paraphrasing.

A number of statistical measures has been proposed to measure how much a candidate word is relevant to the surrounding context. Among different statistical measures for the appropriateness of a candidate substitution to a lexical gap, t-score[3], the conditional probability given the local syntactic context [2], PMI score [4] achieved the best accuracy for the test set provided by Edmonds [3]. As the first method to tackle the problem of context sensitive lexical paraphrasing, Edmonds [3] proposed to use the sum of t-scores of the cooccurrences of a candidate word and context words. Inkpen [4] improved Edmonds' method by using Pointwise Mutual Information (PMI) criteria [5] to measure the associations between a candidate word and the context surrounding the target lexical

gap the system tries to fill in. The appropriateness of a candidate word t to the context is measured by sum of the PMI scores defined as

$$\mathrm{PMI}(w_{-k}^{-1}, w_1^k, t) = \sum_{w \in (w_{-k}^{-1} \cup w_1^k)} \frac{C(w,t)N}{C(w)C(t)}, \tag{1}$$

where w_{-k}^{-1} denotes the k preceding words to the lexical gap, w_1^k denotes the k following words to the lexical gap, $C(\cdot,\cdot)$ denotes the number of the cooccurrences of two words, $C(\cdot)$ denotes the frequency of a word, N denotes the total number of word tokens. Gardiner and Dras [6] presented an approximation of Inkpen's method to accommodate the corpora with provided as N-gram instead of full text.

Bangalore and Ranbow [2] were first to model the task of lexical choice as a multi-class probabilistic classification. They proposed a method to fill a lexical gap by using local syntactic information provided by their syntactic chooser for natural language Generation. In similar task settings, Inkpen significantly improved her unsupervised PMI method with a boosting method with the features of PMI scores and word occurrences in the context windows [4]. Connor and Roth [7] proposed a bootstrapping approach composed of weak learners and a global classifier. A set of weak learners which correspond to context words in a fixed-length window and dependency relations surrounding each lexical gap. A binary classifier with global features, which learns from the aggregated prediction of the weak learners, tries to predict whether the substitution is appropriate or not, given tuple of a original word, a word to substitute with, a context sentence that contains the original word.

It has still not been clear how well lexical choice works for larger vocabulary size, for example, thousands of words or more. In the works that shares Edmonds' experimental setting [3, 4, 6], they only evaluated the performance for specific seven synonym sets composed of the words with less polysemy and similar frequency. The reason why they used such controlled evaluation set was that they wanted to focus on exploring other features than word frequency. Another approach is to treat lexical choice as a binary classification of word substitutions. The existing methods among that type either cover only a specific class of words, such as verbs, in order to exploit local syntactic structure [7], or were evaluated against a human-annotated, but small set of lexical substitutions [8].

We consider that it is promising to apply lexical choice to the term expansion in IR. While traditional term expansion methods without actual user feedback improve the recall of IR systems, it has been well known

that they also tend to invite the risk of degradation of precision [9]. When we have an accurate and wide-coverage lexical choice module, we may achieve improvement of recall with less noisy expansion, by filtering out the expansion candidate which are inappropriate to the surrounding context.

Aiming to the application to term expansion explained above, we focus on constructing models of lexical choice with a larger vocabulary, which provides a more realistic estimation of the effectiveness of lexical choice for the application of IR.

4 PROPOSED METHOD

Our system is composed of two main steps: substitution candidate generator and substitution selector. Given an input document, in the first step, the candidate generator spots the word occurrences which can be substituted, and assigns possible substitution candidates to it, using simple dictionaries of substitutions. In the second step, the substitution selector assigns probabilities to the candidates based on a single maximum entropy model, which allows us to use rich features including document-based features in a single multiclass classifier setting for this problem. We describe each step in detail in the following sections.

4.1 *Substitution candidate generation*

We generate lexical gaps and substitution candidates by simply matching the given document with a *substitution dictionary*. A *substitution dictionary* provides a mapping from a pair of word and its part-of-speech (POS) to the set of its substitutions, which can replace the input word in *some* context. This assumption on dictionaries allows us to exploit an automatically extracted dictionary of substitutions or a domain-specific dictionary curated by human, although we used only a WordNet[10]-derived substitution dictionary in the experiment presented in this paper. Note that, for polysemous words, we simply use a merged set of the words taken from all of their synsets and let the system to choose correct ones considering the context.

4.2 *Substitution Selection*

Given the context C and the set of candidate substitutions S, employing the maximum entropy framework [11], we directly model the conditional

probability of the candidate $w \in S$,

$$P(w|S;C) = \frac{\exp\left[\Lambda \cdot F(w,C,S)\right]}{\sum_{v \in C} \exp\left[\Lambda F(v,C,S)\right]}, \tag{2}$$

where $F = \{f_1, f_2, \cdots, f_K\}$ is a feature vector and Λ is the weight vector. We obtain the most probable substitution candidate $\hat{w}$ for given C and S as

$$\hat{w} = \underset{w \in C}{\operatorname{argmax}}\, P(w|S;C) \tag{3}$$

Here, we choose a point-wise prediction approach, rather than sequential / global optimization approach. Point-wise approaches allows us to try a wide range of features without taking inhibiting computational cost. Instead, it doesn't allow us to directly model the consistency of a combination of predictions.

We will try to capture the consistency as features, as described in the following sections and listed in Table 1. We will discuss limitations of this approach in Section 6.

4.3 *Training*

In order to obtain labeled examples used to train this model, we make a naive assumption that a choice of a word in a given context is correct if and only if the same choice is found in the training corpus. We extract training examples from a POS-tagged corpus by generating lexical gaps and substitution candidates in the same manner as explained in Section 4.1, and assigning positive labels to the actually used words in the training corpus and negative labels to those which is not used, based on the assumption above.

4.4 *Feature extraction*

In order to capture different levels of context in a document, we use three different sets of features: *window-based features document-based features*. In addition that, we also use simple but effective *baseline features*. *window-based* and *baseline* features were essentially the same ones as used in [4].

A *window-based feature* is a function defined on a fixed-length window surrounding the target lexical gap. A *document-based feature* is a function defined on the bag-of-words of the document which contains

the target lexical gap. By using document-based features, we can capture the word associations that occur in long-distance context, including the consistency of the wording across the document.

In syntax-based features, we use the information from the result of syntactic parse of the given sentence. The feature called *subtree* in Table 1, which is one of the syntax-based features, captures the association of a simplified syntactic position and a candidate word. This feature is meant to be beneficial, for example, when the candidate noun has strong tendency to be modified by some adjectives.

For example, from the candidate word *chairman* in the following sentence,

> Mr. Vinken is chairman$_{[p_1]}$ of Elsevier N.V., the Dutch publishing$_{[p_2]}$ group.

we can extract following features.

For the candidate *chairman* in the gap $[p_1]$:

Feature	Value
`frequency`	-10.9
`unigram`*_chairman*	true
`pmi`*_of,chairman*	0.102
`pmiavg`	0.0615
...	

For the candidate *publishing* in the gap $[p_2]$:

Feature	Value
`frequency`	-10.9
`unigram`*_publishing*	true
`pmi`*_Dutch,publishing*	0.102
`pmiavg`	0.0615
`subtree`*_publishing*,`DT/NNP/*VBG/NN`	true
...	

All the features used in the experiment are listed in Table 1.

[4] The value of frequency feature can be real-valued, when we perform normalization.

Table 1. List of features

Feature template	Type	Definition	Value type
`frequency`(w)	baseline	frequency of w	integer [4]
`unigram`_v(w)	baseline	true iff w is v	binary
`pmi`_v(w)	window	$\mathrm{PMI}(v, w)$ for p preceding / q following words	real
`pmiavg`(w,S)	window	$\frac{\sum_{v \in window} \mathrm{PMI}(v, w)}{window\ size}$	real
`cache`_w(w,S)	document	true iff w is seen somewhere else in S except for the original position	binary
`cache-l`_w(w,S)	document	true iff the lemma of w is seen somewhere else in S except for the original position	binary
`cache-o`_w(w,S;C)	document	true iff the lemma of one of C is seen somewhere else in S	binary
`subtree`_v, t(w,S;C)	syntax	true iff the sequence t matches the sequence of the tags of the sibling node of w in S and v	binary
`unigram+pos`_v, T(w,S)	syntax	true iff the pair of surface and POS (w,S) equals to (v,T)	binary

5 EXPERIMENTS

In this section, we compare our all-words maximum entropy model and new document-based features with the Inkpen's supervised method described in Section 3.

5.1 *Experimental settings*

Table 2. Tasks and corpora

		Source corpus	#word tokens
Sample words task	Training	BLLIP 1988 + Penn Treebank	16893445
	Testing	BLLIP 1987	22926540
All words task	Training	Penn Treebank	10778880
	Testing	(4-fold cross validation)	-

Table 3. Part-of-speech mappings

mapping A	mapping B	target POS
NN,NNS,NNP,NNPS	NN	noun
JJ,JJR,JJS	JJ	adjective
VB,VBD, VBG,VBN,VBP,VBZ	VB	verb
others	others	*ignored*

Table 4. Edmonds' seven evaluation sets for lexical choice

Part-of-speech	substitution candidates[5]
adjective	*difficult, tough, hard*
noun	*error, mistake, oversight*
noun	*job, task, duty*
noun	*responsibility, commitment, obligation, burden*
noun	*material, stuff, substance*
verb	*give, provide, offer*
verb	*resolve, settle*

We evaluated our methods with two tasks. The first one, which we call *sample-words* task, is the mostly same experimental setting as [4]. The second one, which we call *all-words* task, is a task in which we try to substitute all of the content words. Table 2 shows statistics of each task. Note that the number of testing samples is very different since we do not limit the substitution candidates to specific set.

In training of both of the two task, we generated substitution candidates using a substitution dictionary that maps a POS-tagged word to the words connected by at least one synonymy relation in WordNet 3.0 [10], which has 117,659 words and 206,941 senses. The parts-of-speech are converted using the mapping A shown in Table 3.

We trained our models with the extracted samples from all sections of the Penn Treebank corpus 3.0 [12] and the sections W8_001 to W8_019 of BLLIP 1987-89 WSJ Corpus. We filtered only the parts-of-speech starting with NN (nouns), JJ (adjectives), VB (verbs) and RB (adverbs) as targets of substitution.

Testing differs for each of the two tasks. In the first setting, which we call *sample-words* task, we compared our method with previous methods of Edmonds' and Inkpen's, by evaluating the prediction accuracy for

[5] Originally Edmonds referred to it as *near-synonyms* [3].

the same corpus, Wall Street Journal of 1987, used by their evaluation. Specifically, we generated lexical gaps and substitution candidates using Edmonds' seven near-synonym sets shown in Table 4 for evaluation, which is shared by [3], [4] and [6].

In the second setting, which we call *all-words* task, we tried to substitute all the content words in the given text. The intention behind this setting is that we want to evaluate for a wider-coverage of substitution candidates, aiming to applying lexical substitution to term expansion in IR.

We evaluated our models with the extracted samples from the sections W7_001 to W7_127 of BLLIP 1987-89 WSJ Corpus. PMI scores are estimated on the sections from W8_001 to W8_108 and from W9_001 to W9_41 of BLLIP WSJ Corpus, with the 5 words window and normalization with a sigmoid function. Note that the corpus used to estimate PMI was a relatively small corpus consists of domain specific texts.

For preprosessing the queries to substitution dictionaries and for the `cache-l` feature explained in Table 1, we lemmatized the target words by using the rewriting rules and exception lists provided in the WordNet implementation. For the parameter of `pmi` features, we used $p = 3$ and $q = 2$.

Since Inkpen did not explicitly mention about how she obtained and used part-of-speech information to identify target gaps to fill, we could not make faithful reproduction of her experimental settings. Gardiner [6] also reported similar difficulty in reproducing the settings. As a result, the sample-words task is slightly different from Inkpen's task. This difference can be seen in the difference between the numbers of the test samples. We assume that this difference comes from the different mappings from fine-grained parts-of-speech in the corpus to WordNet's coarse-grained ones. Table 3 shows two criteria we used to map from the tag set of Penn Treebank to three coarse parts-of-speech, namely, noun, adjective and verb, which are used in WordNet.

5.2 *Evaluation method*

Since the cost of human judgements for the size of corpora given in Table 2 is prohibitive, we evaluate the results with the exact match to the original word in a gap. This evaluation also serves as a mean to compare our method with previous researches, since this was the one used by the state-of-the-art method of Inkpen's [4]. As Inkpen mentioned, the exact match gives substantial underestimation of the true accuracy, since some of the

substitution candidates other than the original one can be acceptable in the given context.

5.3 *Experimental results*

Table 5.3 shows the experimental results of Sample-words task. Table 5.3 shows the experimental results of All-words task. Every difference between a pair of accuracy values listed in the tables is shown to be statistically significant using McNemar's test with $p < 0.05$.

Table 5.3 shows the performance comparison of the proposed method (Proposed-A, Proposed-B) with the state-of-the-art method of Inkpen's (Inkpen). In this result, we used the setting of sample-words of Table 2 to obtain the almost same experimental condition as Inkpen's. The suffix "-A" or "-B" denotes the part-of-speech mapping chosen from the two in Table 3. The suffix "+Doc" denotes that we used document-based features in addition to baseline and window features given in Table 1. Similarly, "+Syn" denotes the incorporation of syntax-based features.

The proposed method outperformed the state-of-the-art of Inkpen's against the test sets of nouns and adjectives. Both of Proposed A and Proposed-B make the use of the essentially same feature set as the Inkpen's method. We suppose that the improvement came from the fact that we used a domain-specific corpus when PMI scores are calculated, whereas Inkpen used web derived data of general domain.

Furthermore, the incorporation of document-based features slightly improved the performance of the proposed method in nouns and adjectives. However, the same incorporation was not effective for the verbs in the test set of sample-words.

In all-words task, in contrast to sample-words task, the incorporation of both of document-based and syntax-based features improved the accuracy significantly.

6 DISCUSSION AND FUTURE DIRECTIONS

6.1 *Differences between Sample-words and All-words*

Our models showed substantially different results between sample-words and all-words tasks. An important difference was that syntax-based features improved the results of all-words task. We suppose that this is caused by the difference of quality of candidate sets. In fact, in sample-words

Table 5. Accuracy for sample-words task on BLLIP corpus of Wall Street Journal 1987

	Acc. (noun and adj.)	Acc. (all)
Inkpen	70.01	65.16
F0	70.01	65.16
A	70.95	58.29
B	72.03	65.03
A +Doc	71.09	58.18
B +Doc	72.19	64.63
A +Doc+Syn	65.96	50.73
B +Doc+Syn	64.13	45.49

Table 6. Accuracy for All-words task on Penn Treebank

	Acc. (noun and adj.)	Acc. (all)	#samples (noun and adj.)	#samples
A	79.23	75.07	86179	122131
A +Doc	81.20	77.00	86179	122131
A +Doc+Syn	81.59	77.47	86179	122131

task, candidate substitutions has really similar usages including subcategorization of arguments. Similar claim has been mentioned in [4], where it was claimed that Edmonds' sets were close enough to WordNet synsets. In contrast to that, candidate substitutions in all-words task are more diverse including strong polysemy that can have clearly separated word usage associated with the context. We suppose that subtle distinction of among the sets like WordNet synsets may not lead to benefit of application tasks including term expansion, while the effectiveness of syntax-based and document-based features in all-words is a promising result towards such applications.

6.2 *Contribution of document-based features*

Document-based features improved the performance both in all-words and sample-words tasks. This suggests that it is important to catch the consistency of terminology in choosing a word from synonymous candidates.

However, we should be careful about the fact that we took point-wise prediction strategy, which means that in the selection for gap, we assumed correct predictions in other gaps. In real applications, we may want to perform lexical choice for different gaps at the same time. In such cases, point-wise assumption may not be appropriate.

6.3 *Lexical variation and ambiguity*

People use different terminologies to tell the same thing, according to the writer's and/or reader's background. This fact suffers IR systems through two ways; lexical variation and lexical ambiguity. Lexical variation in natural language texts, including spelling variation, abbreviation and aliases, makes it difficult for IR systems to find the relevance between queries and documents. Lexical ambiguity, which sometimes cooccurs with lexical variation, is the problem that a term can refer to more than one thing. A typical example of lexical ambiguity is the word *pitch*, which can refer to a throw of a ball, to the property that describes the frequency of sound, or to a sticky substance secreted by trees. There is a need for disambiguating ambiguous terms in queries or documents, since they can cause wrong association between queries in IR systems.

A common approach to solving lexical variation is the expansion of terms used in documents and queries. In expansion approaches to lexical variation, there is a risk of introducing noise in inappropriately expanded terms via a non-relevant sense to the context.

To solve lexical ambiguity and lexical variation at the same time, many researchers have been trying to effectively apply word sense disambiguation (WSD) to IR. In the early attempts in applying WSD to IR, while researchers found there was ambiguous terms in real environment of IR, they achieved little improvement [13]. By introducing artificially created pseudo ambiguous words, Sanderson analyzed the effectiveness of WSD, changing the query length and the accuracy of automatic WSD [14]. He found that, as for query length, WSD was more effective on queries with less terms, such as one or two; as for WSD accuracy, he concluded that at least 90% accuracy is required to make improvements in IR performance.

An inherent limitation of WSD-based approaches to lexical variation is that it is not clear what level of granularity is most appropriate for IR. It may be even better to have different level of granularity according to the domain or topic of specific applications, than to have a single consistent criteria. For example, WordNet[10] version 3.0 gives two distinct

senses to the word *Atlanta*. One refers to the capital of Georgia in the United States of America. The other refers to a historic battle during the American Civil War. On the other hand, Wikipedia provides more than thirty referents including eighteen place names and five ship names, for the word *Atlanta* [6].

Lexical substitution has been studied as a "relaxed" version of word sense disambiguation, in which one do not prepare a predefined set of senses, or sense inventory. In the approach of lexical substitution, instead of having sense inventories, one tries to find a set of possible substitution for a word and select the best candidate given a specific context of the word token [15].

6.4 *Remaining problems*

The most important part missing in this work is finer evaluation of our method. Our evaluation framework is not as accurate as those used in word sense disambiguation or paraphrasing, in which human annotated corpora or human judgement is used. Since our evaluation measure misjudges some correct answers as errors, as discussed in Section 6, the effectiveness of document-based and syntax-based features should be further investigated in the evaluation framework with human judgment such as lexical substitution tasks [15].

Meanwhile, we also can evaluate our method in terms of performance on real-world applications including machine translation and information retrieval. Real-world applications might not be affected by the noise in the training data, especially when the size of training data is large.

7 CONCLUSION

We have seen that in our all-words task that has larger vocabulary of candidate substitutions, rich features—including document-based and syntax-based features—improved the performance of context sensitive lexical choice.

---

[6] Atlanta (disambiguation). (2009, August 17). In *Wikipedia, the free encyclopedia*. Retrieved on 13:47, October 1, 2009, from http://en.wikipedia.org/w/index.php?title=Atlanta_(disambiguation)&oldid=308397379.

REFERENCES

1. Yamamoto, K.: Machine translation by interaction between paraphraser and transfer. Proceedings of the 19th International Conference on Computational Linguistics (COLING), 2002 (2002)
2. Bangalore, S., Rambow, O.: Corpus-based lexical choice in natural language generation. In: ACL. (2000)
3. Edmonds, P.: Choosing the word most typical in context using a lexical co-occurrence network. In: Proceedings of the Europiean Association of Computational Linguistics. (1997)
4. Inkpen, D.: A statistical model for near-synonym choice. ACM Trans. Speech Lang. Process. **4**(1) (2007) 2
5. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. **16**(1) (1990) 22–29
6. Gardiner, M., Dras, M.: Exploring approaches to discriminating among near-synonyms. In: Proceedings of the Australasian Language Technology Workshop. (2007)
7. Connor, M., Roth, D.: Context sensitive paraphrasing with a global unsupervised classifier. In: Proceedings of the 18th European conference on Machine Learning, Berlin, Heidelberg, Springer-Verlag (2007) 104–115
8. McCarthy, D., Navigli, R.: Semeval-2007 task 10: English lexical substitution task. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, Association for Computational Linguistics (June 2007) 48–53
9. Manning, C.D., Raghavan, P., Schtze, H.: Relevance feedback and query expansion. In: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
10. Fellbaum, C.: WordNet: An Electornic Lexical Database. Bradford Books (1998)
11. Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A maximum entropy approach to natural language processing. Comput. Linguist. **22**(1) (1996) 39–71
12. Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn Treebank: annotating predicate argument structure. In: HLT '94: Proceedings of the workshop on Human Language Technology, Morristown, NJ, USA, Association for Computational Linguistics (1994) 114–119
13. Krovetz, R., Croft, W.B.: Lexical ambiguity and information retrieval. ACM Trans. Inf. Syst. **10**(2) (1992) 115–141
14. Sanderson, M.: Word sense disambiguation and information retrieval. In: SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, Springer-Verlag New York, Inc. (1994) 142–151
15. McCarthy, D., Navigli, R.: The English lexical substitution task. Language Resources and Evaluation (2009)

YUSUKE MATSUBARA
DEPARTMENT OF COMPUTER SCIENCE,
THE UNIVERSITY OF TOKYO,
7-3-1 HONGO, BUNKYO-KU, TOKYO,
JAPAN
E-MAIL: <MATUBARA@IS.S.U-TOKYO.AC.JP>

JUN'ICHI TSUJII
DEPARTMENT OF COMPUTER SCIENCE,
THE UNIVERSITY OF TOKYO,
7-3-1 HONGO, BUNKYO-KU, TOKYO,
JAPAN
AND
NATIONAL CENTRE FOR TEXT MINING,
MANCHESTER INTERDISCIPLINARY BIOCENTRE,
131 PRINCESS STREET, MANCHESTER,
UK
E-MAIL: <TSUJII@IS.S.U-TOKYO.AC.JP>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 25–40
RECEIVED 01/11/10 ACCEPTED 21/12/10 FINAL 08/02/11

Folding FrameNet – Unfolding Its Potential?

JOSEF RUPPENHOFER AND MANFRED PINKAL

Saarland University, Germany

ABSTRACT

FrameNet holds intuitive appeal as a resource that provides valuable semantic information for NLP systems, but its impact on NLP applications consistently falls short of expectations, due to the poor performance of frame-based semantic parsers. Sparse data and high granularity have been identified as interrelated reasons. In this paper, we study the effect of coarsening FrameNet, using a tool for automatic merging of frames and lexical units. We report results of experiments carried out to assess the effect of granularity reduction on parsing performance and on the discriminative power of frame information for the RTE task. A qualitative examination of annotation changes affected by the merging process leads to interesting conclusions concerning general desiderata in semantic processing and in FN-based lexical-semantic modeling, and on the general usefulness of granularity reduction.

1 INTRODUCTION

Predicate-argument structure has become a central part of information extraction, question answering and other information access tasks [1–3]. The most widely used resources for modeling predicate-argument structure in English are PropBank [4] and FrameNet [5]. PropBank abstracts over variations in the syntactic realization of an individual verb's semantic roles. PropBanks are available for an expanding set of languages, they can be created efficiently and have been integrated into a number of NLP systems [6]. PropBank analysis is, however, rather syntax-dependent, and it does not generalize across lemma boundaries. FrameNet (henceforth

also: FN) provides such generalization by grouping relevant senses of different lemmas into the same frame with a shared set of semantic roles, called Frame Elements. Moreover, FN organizes both frames and frame elements into a hierarchy using several types of semantic relations. Via its frame-to-frame relations, FN also models presuppositions of events and generates expectations in the sense that event types (frames) are connected through sequential and other relations. Finally, frame structures are language-independent, and thus suggest themselves for cross-lingual applications.

We use an example from the RTE-2 challenge [7] to illustrate FN's contribution to NLP applications. The RTE task consists in determining whether a text entails a hypothesis in a common-sense way. It uses data from IR, IE, QA and Summarization to form positive and negative text-hypothesis pairs.

(1) T:. : Mr. Fitzgerald revealed he was one of several top officials who told Mr. Libby in June 2003 that Valerie Plame, **wife** of the former ambassador Joseph Wilson, worked for the CIA.
H:. Valerie Plame is **married** to Joseph Wilson.

Frame semantic analysis with a well-performing shallow semantic parser would show *wife* in the text and *married* in the hypothesis to both belong to the `Personal relationship` frame, constituting evidence that entailment holds.

Despite the intuitive appeal of FN, its impact on NLP applications falls short of expectations. An often-mentioned practical drawback is the lack of coverage, due to the incompleteness of the FN database [8]. Even more importantly, it is very difficult to leverage the information which is covered by the FN database for practical tasks [9]. Existing FN-based shallow semantic parsers show consistent low performance, in contrast to the rather impressive accuracy of PropBank role labelers. Of course, the task of assigning frame structures is more ambitious and thus simply harder. But another reason may be that FN is just too fine-grained to allow robust shallow semantic parsing. On the one hand, the high granularity increases the problem of data sparseness: for many frames, frame elements and lexical units, the FN corpus contains only few annotated training instances. On the other hand, it seems that semantic distinctions often are too subtle for parsers – and for humans.

In this paper, we study the effect of systematically coarsening FN. This directly addresses problems of granularity and indirectly issues of coverage and sparsity: the merging of senses provides more annotated

instances for the remaining word senses and may eliminate word senses that are defined by FN but not backed by annotations. We use the recently presented FN transformer, which automatically merges frames or word senses and changes annotations of corpus data according to user specifications [10], to systematically vary the granularity of FN, and evaluate the effect of frame folding on parsing accuracy. By themselves, effects on parser performance are of limited interest since frame folding implies a reduction of alternative target annotations, almost trivially entailing an increase in accuracy. The interesting question is this: Does parser improvement come at the cost of losing relevant semantic information? Or could we even gain information through additional cases of semantic relatedness becoming apparent through the merging process? To answer these questions, we build on the experimental setup of Burchardt et al. 2009 [9] for evaluating the impact of FN on the task of Recognizing Textual Entailment (RTE).

These authors use the FATE corpus [11], which contains manual frame semantic annotations for the RTE-2 test set, as a gold corpus. Unlike end-to-end evaluations, this setup allows them to separately measure FN's coverage, the performance of semantic parsers, and the discriminative power of frame-semantic information for entailment recognition. Concerning the latter, they find a discriminative effect of frame information on the entailment task which significantly exceeds a simple word-overlap measure. However, this positive effect can only be measured on the gold corpus; in a realistic setting it is offset by noisy parser output. We replicate their experiments in a somewhat modified version with FNs of different granularity to assess whether coarsening can lead to better automatic parser performance *while avoiding the loss of relevant information.*

Our evaluation shows that parsing results improve significantly by merging, but there is no significant gain in discriminative power. We complement the quantitative evaluation by a qualitative examination of those entailment pairs which are affected by the merging process. This leads to interesting conclusions concerning general desiderata in semantic processing and in FN-based lexical-semantic modeling, and on the general usefulness of frame-folding.

This paper is structured as follows. In section 2, we discuss related work. In section 3, we report on experiments in which we systematically explore the coarsening of FN with the FN transformer tool and its effect on the performance of a statistical semantic parser. In section 4, we use the output of the best transformer setting and replicate parts of Burchardt et al.'s study (2009) on the impact of frame semantic information on the

RTE task. In section 5, we present a qualitative evaluation of the effects of collapsing frame and lexical unit distinctions. In section 6, we offer our conclusions.

2 Related Work

McConville and Dzikovska [12] pursue an approach aimed at reducing FrameNet's granularity. They build a new semantic resource by deriving a verb lexicon for deep syntactic parsing from the annotations in FrameNet using the `Inheritance` relation between frames and the `CoreSet` relation between Frame Elements to reduce the set of semantic roles. Their lexicon comes, however, without an appropriately updated corpus. Therefore, it cannot be used to train a shallow semantic parser. Ruppenhofer et al.'s [10] transformer tool, which we will use in this paper, was specifically created to allow for experimenting with different levels of granularity in FrameNet to suit NLP applications.

Matsubayashi et al. [13] focus on the sparse-data problem for role assignment. They compare various ways of generalizing across semantic roles. Fürstenau and Lapata [14] propose a semi-supervised approach to augment the training material for verbs known to FrameNet based on small seed sets. The performance effects on a semantic role labeler are limited and they are best for smaller seed sets. In another paper, Fürstenau and Lapata [15] propose a semi-supervised approach to assign instances of verbs that are unknown to FrameNet to suitable frames. Further coverage-related work concerns type-based lexical unit induction. Pennacchiotti et al. [16] use vector space models and WordNet relations to assign unknown predicates to the most similar frame.

3 Parsing Experiments

In this section, we discuss how some of FrameNet's frame and word sense distinctions can be collapsed in a linguistically motivated way, and measure the effect on the performance of a shallow semantic parser. To this end, we systematically explore different parameter settings for the FrameNet transformer tool to produce modified versions of the FrameNet data. We then train and test a statistical semantic parser on the modified FrameNet data to evaluate its performance.

We use FrameNet's official release 1.3 and modified versions of it that we generate using the transformer tool. FrameNet release 1.3 has 795

frames and 10195 lexical units (word senses), belonging to 8365 different lemmas. 6728 lexical units are exemplified with annotated instances that can be used as training data.[1]

3.1 *Frame Folding*

The FrameNet Transformer tool gives users a possibility to automatically coarsen the FrameNet sense inventory in an iterative procedure. The tool automatically merges either entire frames, if they stand in appropriate frame relations chosen by the user, or word senses of lemmas that belong to frames related by the specified relations. In the latter mode, only word senses standing in a child-ancestor relation are merged. In either mode, the transformer automatically outputs format-compliant FrameNet versions, including modified corpus annotation files that can be used for automatic processing. Users can vary the behavior of the FN transformer by setting paramaters. For our experiment, we selected eight linguistically reasonable parameter combinations. We describe and motivate them in the following.

Type of merger. Frame-based merging aims at reduction of frame granularity and providing larger sets of training data per frame. Lexical-unit based merging aims at reduction of word-sense ambiguity, and leaves the frame structure unchanged. We tested frame-based and LU-based merging, as well as combinations of the two in either order. In Table 1 below, we use F to designate a frame based merger, L to designate lemma-based one, FL for a sequence of frame-based merging followed by LU-based merging, and LF for a sequence of mergers in the reverse order.

Frame relations licensing merge. The FrameNet database employs different frame-to-frame relations, a subset of which can be specified by the user of the transformer tool. In the frame-based mode, the transformer merges those pairs of frames which are connected by licensing relations. In the LU-based mode, the transformer traverses licensing relations downward from the frame of the target LU to find lemma-identical LUs to merge into the target LU. E.g., with the lemma *depart*, the sense in the frame `Departing` can serve as a target for the sense in the child frame `Quitting a place`, which is related by INHERITANCE. The PERSPECTIVE_ON, SUBFRAME, INCHOATIVE_OF, and CAUSATIVE_OF relations are reliable indicators of close semantic relatedness. We use

[1] Very recently, FrameNet made available a new release, 1.5. Its format is different from release 1.3 and therefore not compatible with the FrameNet transformer tool we use.

them throughout in our experiments. We exclude the USING relation because it often connects frames that are only vaguely related. INHERITANCE is not a clear positive candidate because the semantic step length between frames connected by this relation varies greatly. We carry out every experiment in two versions, one with and one without INHERITANCE as a licensing relation. In the Setting column of table 1, we refer to these versions as F+I, F-I, etc.

Stop frames. Stop frames are frames which may not serve as target frames for frame-based mergers. Stop frames are highly abstract frames like Event, Process, Transitive_action, which are connected to large numbers of very different frames that we do not want to collapse. We use the same set of 12 stop frames in all experiments. Stop frames are used only in frame-based merging.

Number of iterations performed. The FN transformer proceeds iteratively, always merging (lexical units in) adjacent frames. We set the number of iterations to 3 for frame-based merging, and 1 for LU-based merging because only few additional mergers take place if one performs additional iterations.

We use eight different parameter combinations, each starting from the official FrameNet release and producing a different modified FrameNet version. These parameterizations share the appropriate fixed settings but vary with regard to merger type and the use of the INHERITANCE relation in identifying source frames and LUs.

3.2 *Parsing*

To automatically annotate the nine (original and derived) versions of FN, we used the Shalmaneser shallow semantic parser [17], which was the only freely available parser for frame semantic role labeling at the time.[2] Shalmaneser breaks down the task of frame semantic annotation into three ordered sub-problems (frame assignment, argument recognition, and argument labeling) which are modeled as modular, supervised learning tasks. We used the default Naive Bayes classifiers that come with Shalmaneser.

We trained and tested the Shalmaneser semantic parser in a 10-fold cross validation setting. We trained each system only on those lemmas that were affected in at least one of our eight experiments, because only in

[2] The system Johansson and Nugues built for the Semeval-2007 task was no longer available [18]. CMU's SEMAFOR-system [19] became available too late for use in this work.

Table 1. Performance of SRL system trained on different versions of FN

ID	Setting	LU red.	avg. # senses	frame assign.	arg. recognition			arglab
				acc.	prec.	rec.	f-score	acc.
-	ORIG	0	1.66	0.938	0.852	0.697	0.766	0.762
1	F, +I	153	1.55	0.944	0.846	0.682	0.755	0.746
2	**LF, +I**	**397**	**1.36**	**0.963**	**0.842**	**0.674**	**0.749**	**0.741**
3	L, +I	353	1.39	0.958	0.845	0.691	0.760	0.746
4	FL,+I	369	1.38	0.956	0.843	0.676	0.750	0.738
5	F, -I	112	1.58	0.943	0.848	0.680	0.755	0.750
6	LF, -I	339	1.40	0.958	0.846	0.677	0.752	0.742
7	L, -I	352	1.39	0.959	0.848	0.692	0.762	0.748
8	FL, -I	360	1.39	0.957	0.849	0.683	0.757	0.745

these cases the parser's behavior may change. The data set contains 1300 lemmas and 36681 annotated frame instances (annotation sets). The 1300 lemmas involved have a total of 2163 word senses associated with them in the official FrameNet release. 755 of the lemmas are unambiguous. They are included in the data set because their single sense was affected by a frame-based merger. 354 lemmas have 2 senses and 191 lemmas have more than 2 senses. The reduction of senses in the sub-corpus under consideration can be read off the third column of Table 1.

3.3 *Results*

The results of our experiments, shown in Table 1, confirm that we can improve parser performance by collapsing certain distinctions made by FrameNet. Not too surprisingly, frame assignment accuracy correlates with the degree of frame ambiguity. The best performance on frame assignment results from experiment 2, where polysemy was reduced the most, by combining LU-based merging with frame-based merging in that order, and allowing INHERITANCE. Table 1 further shows that, unlike with frame assignment, the modified FN versions cannot outperform the original release when it comes to argument recognition and labeling.

4 RTE EXPERIMENTS

In our second experiment, we turn to the question how frame folding affects the relevant semantic information contained in frame annotations.

We draw on the work of Burchardt et al.(2009), who made a study of the impact of frame semantic information on the RTE task. Like them, we base our study on the FATE corpus.

4.1 *The FATE Corpus*

The FATE corpus [11] contains manual annotations of the RTE-2 test set, consisting of 800 entailment pairs, 400 positive, 400 negative. In positive entailment pairs, the sentences are not fully annotated with semantic frames but only on relevant spans which annotators deemed to have a bearing on making the entailment decision. The FATE corpus contains 1686 sentences with 4490 annotated frame instances and 9518 role instances.

Lemmas in FATE were annotated with frames in a rather flexible way. For instance, annotators were allowed to apply frames to occurrences of lemmas if the frames seemed to match the occurring word senses, even if the lemmas were not listed in the relevant frames in FrameNet. Due to this generous annotation policy in FATE, Burchardt et al. report a surprisingly high coverage of 92% on frame instances. If we only consider annotated instances of frames for which the frame-evoking lemma is listed in FN 1.3 and has training data, we are left with a total of 1519 frame instances (34%) that the Shalmaneser parser can possibly handle.[3] We call this subset of FATE annotations FATE-strict.

To fairly assess the performance of the Shalmaneser semantic parser, we use FATE-strict as a gold standard. We train the parser on FrameNet release 1.3 and test it on FATE-strict. On the 1519 annotatable instances in FATE-strict, the system achieves a precision of 86% and a recall of close to 100% for frame assignment. Accordingly, accuracy also stands at close to 86%, which is somewhat lower than the 93% accuracy value we obtained when training and testing with Shalmaneser on FrameNet release 1.3. The precision and recall figures we obtained in our experiment contrast markedly with the precision and recall values of 0.35 and 0.40 for Shalmaneser given by Burchardt et al., who evaluated Shalmaneser against the full set of FATE annotations, including the cases that Shalmaneser was not equipped to handle.

[3] The Shalmaneser semantic parser can only assign a frame to a lemma instance if it has seen training data for that lemma-frame pair. It cannot transfer what it has learned on a lemma with training data for a given frame, to another lemma without training data.

4.2 *Experimental Setup*

We follow Burchardt et al 2009 in using the FATE corpus but we will not attempt to replicate their experiments in full. The focus of our analysis is on comparing versions of the FATE corpus which are annotated according to frame schemas reflecting different levels of granularity.

The basic method is to extract frame-based statistical information from the positive and negative entailment pairs in the annotated corpus, respectively, and to measure the overlap of frame structures between the text and the hypothesis sentences. The key assumption behind this method is that the more of the semantic material in the hypothesis can be 'embedded' into the text, the more likely it is that an entailment relation exists between text and hypothesis. For the level of frames or word senses, Burchardt et al. define a **frame label overlap** measure between hypothesis sentences and the text sentences paired with them. Frame label overlap is defined as the percentage of frame labels in the hypothesis which have a counterpart in the text. Frame label overlap is calculated separately for the frames in all positive and negative pairs. The difference between these two scores is taken as a measure of the discriminative power contributed by using frame semantic information.

Our major purpose is to compare the effect of frame folding on frame label overlap information and difference values. We decided to only use the maximally reduced FN version, produced by parsing experiment 2, and compare it to the original FrameNet annotation because it leads to the highest parser quality and provides us with the greatest amount of possible frame assignment differences. We compute the overlap and difference scores for three versions of the corpus,

- the original generously annotated version of FATE, which we use as a gold corpus; the scores give us information about how optimally available FN information (at the original and reduced level of granularity) will be able to discriminate positive and negative entailment cases
- FATE-strict, i.e., the corpus constrained to the parsable frame instances; this is not very interesting in itself, but is needed for comparison with the third version:
- the constrained version of the corpus annotated by the Shalmaneser system; the scores for this version are an indicator of how much frame information can be accessed in a realistic setting.

4.3 *Results*

Table 2 gives the detailed results of the experiment. In addition to the frame overlap scores for positive and negative entailment pairs, and the difference between the two, it shows the corresponding values for word overlap measured on the sets of annotated instances in the generous and the constrained corpus versions.

The results show that for all versions of FATE, frame label overlap scores are higher than lemma overlap scores, underlining the fact that frame semantic normalization brings out latent semantic overlap that is not captured by lemma overlap alone. Also, we could replicate the result of Burchardt et al. that the difference in frame overlap outperforms word overlap by 3.5%. Moreover, we achieve consistently higher frame label overlap scores for the merged versions than on the corresponding versions annotated according to the original FrameNet scheme. This is as expected since the modified scheme reduces the number of frames. However, unlike what we had hoped for, we find that frame label overlap not only increases on positive entailment pairs but also on negative ones. In fact, for all three of the merged versions, there is a small decrease compared to their full FrameNet counterpart versions.

Overall, the quantitative results seem to paint a disappointing picture for the possible contribution that the collapsing of frame and lexical unit distinctions could make to the task of entailment recognition. However, before we accept such a general assessment we will take a closer look at the corpus data themselves.

Table 2. Frame label and lemma overlap on entailment pairs

	Pos	Neg	Diff
Gold Corpus Generous			
Original FN	0.571	0.459	0.112
Merged	0.575	0.490	0.085
Lexical Overlap Baseline	0.450	0.373	0.076
Gold Corpus Constrained			
Original FN	0.549	0.453	0.096
Merged	0.570	0.480	0.090
Lexical Overlap Baseline	0.525	0.410	0.115
Shalmaneser			
Original FN	0.524	0.433	0.091
Merged	0.530	0.450	0.080

5 QUALITATIVE STUDY

In order to get a better sense of the effect of collapsing frame and lexical unit distinctions, we will examine the differences between original and merged FrameNet annotation in detail. We will focus on the generous version of the gold corpus, to have a maximum amount of clean data available. Actually, the subset of text-hypothesis pairs seeing changes in their overlap score is rather small: 49 pairs out of 774 (6.3%). However, note that 541 of the 774 pairs have at least one changed frame instance in the transposed FATE (69.9%). Based on manual inspection of 50 randomly chosen pairs with changing annotation and unchanged score, we find that in the majority of cases (about 80%), a frame that is being shifted occurs only on the T(ext) or only on the H(ypothesis) side, its shift thus having no impact on the overlap score. In the remaining 20% of cases, the same frame is changed in identical ways on both T and H.

We now focus on the 49 entailment pairs where relevant changes occur. In 17 cases, the change that occurs works in our favor. 11 cases are textbook examples of what we hoped to achieve by folding frames: different lemmas and frames on positive pairs in the original version of the corpus come to be aligned in the transformed corpus. An example of this is (2), where *accuse* shifts from `Notification_of_charges` to `Crime_scenario` on the H side, while *arrest* and *try* on the T side shift to the same frame from `Arrest` and `Trial`, respectively.

(2) T: Wyniemko , now 54 and living in Rochester Hills , was **arrested** and **tried** in 1994 for a **rape** in Clinton Township.
H: Wyniemko was **accused** of rape .

In the other 6 cases, frame instances in negative pairs that are aligned in the original FATE come apart, yielding lower overlap scores. However, under ideal conditions none of these changes would have happened, and the lack of entailment would have been recognized by another module of an RTE system. 4 of these cases are due preprocessing errors in the FATE corpus (incorrect lemmatization). 1 case is the result of a coverage gap in FrameNet and the two lemmas in question both should have moved to the same frame. The final case involves two different lemmas on the text and hypothesis side, which do not share the same kind of polysemy and thus cannot move to the same frame during the lexical-unit based second merger phase of parsing experiment 2. The relevant case is (3), where *hang* and *execute* both have senses in the `Killing` frame but only

execute has a sense in the `Intentionally_act` frame, with which its `Killing` sense can merge in the LU-based merger step.

(3) T: Some 420 people have been **hanged** in Singapore since 1991 , mostly for drug trafficking , an Amnesty International 2004 report said . That gives the country of 4.4 million people the highest execution rate in the world relative to population .
H: 4.4 million people were **executed** in Singapore .

In 32 of the 49 pairs with changes in the overlap score, the change runs against our interest. In 9 positive entailment pairs, two correctly aligned frame instances come apart during the lexical unit-based second merger phase because they have two different lemmas that do not share the same polysemy, similarly to what happens on the negative pair exemplified in (3). In another case, two correctly aligned frame instances come apart because of a lemmatization error. In the remaining 22 cases, all in negative entailment pairs, two frame instances are correctly brought into alignment in the course of frame-based merging. The lack of entailment in most of these cases depends on aspects which are out of the scope of lexical semantics. In some cases, the compositional process brings targets bearing identical frame information into completely different contexts. For instance, in example (4), the *Entity* frame elements of the two frame instances do not refer to the same entities. In other cases, modal operators or negation influence the veridicality of a predicate-argument structure and thereby prevent entailment. In (7), the embedding of the `Forming_relationships` frame evoked by *married* under the modal operator introduced by *possibility* prevents entailment. Example (5) is a different case. Here, the opposed scalar values of *cut* and *rise* are not differentiated in FrameNet, although the opposite polarity is a matter of lexical semantics.

(4) T: **Changes** [in the cell-cycle and apoptotic machineries , or in the signaling pathways that control them ^{Entity}] allow cancer cells to escape the normal control of cell proliferation and cell death .
H: The **altered** [cellular networks of molecular pathways ^{Entity}] sustain cancer cell growth and make them resistant to certain therapies .

(5) T: The National Institute for Health and Clinical Excellence estimates the move would **cut** the number of unplanned pregnancies by 70,000 each year .
H: Unplanned pregnancies **rise** by 70,000 each year .

(6) Still , violence continued : Insurgents killed five U . S . soldiers , set off a suicide car bomb that **killed** [four Iraqi policemen ^{Victim}] in Baghdad and targeted more polling sites across the country .
H: Five U S soldiers were killed , and at least [10 Iraqis ^{Protagonist}] **died** in Baghdad

(7) T: The possibility for Yevgenia Timoshenko , daughter of the Ukrainian ex-prime minister , to be **married** to the English rock singer Sean Carr , in church , is still questionable
H: Yevgenia Timoshenko is the **wife** of Sean Carr .

Overall, our examination of the entailment pairs with changed overlap scores suggests that collapsing frame distinctions is useful, if we look past the simple metric of frame overlap scores. First, the predominantly negative effects on merging by pre-processing errors can be set aside as orthogonal issues. More importantly, while the easy cases of frame instances coming into correct alignment to increase overlap on positive pairs are an argument for the merging, so are the cases of frame instances coming into correct alignment on negative pairs (5-7). For these latter cases, too, it is crucial that they come into alignment since other RTE modules checking on correct semantic composition, quantification etc presuppose that the relations they are operating on are correctly identified as equivalent.

The handling of antonyms could readily be improved by augmenting FrameNet's semantic representation. Antonyms should either be handled by separating them into distinct frames or recording their scalar properties on each lexical unit. Finally, based on this data set, lexical-unit based merging seems to mostly have ill effects, often breaking up correct alignments on positive pairs. With this type of merger, the benefit of improved parsing accuracy in the abstract seems to be outweighed by the loss of relevant information in the RTE setting.

6 CONCLUSION

We investigated the effect of automatic coarsening of FrameNet on the performance of a semantic parser, and on the usefulness of frame-semantic information for NLP tasks, using the example of RTE. We have shown that coarsening FrameNet improves performance on the frame assignment task but incurs a slight drop on the role recognition and labeling tasks. The quantitative evaluation of the impact of frame-folding on the RTE task did not show a positive result. A qualitative study of the induced

changes in the annotation gave a differentiated, but basically positive picture. Frame-merging is responsible for most changes, they almost always lead to locally correct alignments, but must be complemented with mechanisms to handle non-local operators and compositional structure. LU-based merging seems to be less advisable because it often has negative effects. The FrameNet database should not only be completed, but extended to include further layers of lexical-semantic information such as polarity.

ACKNOWLEDGMENTS This work was funded by the German Research Foundation DFG (PI 154/9-3).

REFERENCES

1. Kulick, S., Bies, A., Liberman, M., Mandel, M., McDonald, R., Palmer, M., Schein, A., Ungar, L.: Integrated annotation for biomedical information extraction. In: HLT-NAACL 2004 Workshop: BioLINK 2004, Linking Biological Literature, Ontologies and Databases, Boston, ACL (2004) 61–68
2. Kaisser, M., Webber, B.: Question answering based on semantic roles. In: Proceedings of the Workshop on Deep Linguistic Processing, Prague, ACL (2007) 41–48
3. Fliedner, G.: Linguistically Informed Question Answering. Ph.d., Saarland University, Saarbrücken (2007)
4. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: A Corpus Annotated with Semantic Roles. Computational Linguistics **31**(1) (2005) 71–106
5. Baker, C., Fillmore, C., Lowe, J.: The Berkeley FrameNet Project. In: Proceedings of the 17th International Conference on Computational Linguistics, Montreal, ACL (1998) 86–90
6. Carreras, X., Màrquez, L.: Introduction to the conll-2005 shared task: semantic role labeling. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, Ann Arbor, ACL (2005) 152–164
7. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice (2006)
8. Palmer, A., Sporleder, C.: Evaluating framenet-style semantic parsing: the role of coverage gaps in framenet. In: Coling 2010: Posters, Beijing, Coling 2010 Organizing Committee (2010) 928–936
9. Burchardt, A., Pennacchiotti, M., Thater, S., Pinkal, M.: Assessing the Impact of Frame Semantics on Textual Entailment. Journal of Natural Language Engineering **15**(4) (2009) 527–550

10. Ruppenhofer, J., Sunde, J., Pinkal, M.: Generating FrameNets of various granularities: The FrameNet Transformer. In: Proceedings of the Seventh conference on International Language Resources and Evaluation, La Valletta, ELRA (2010)
11. Burchardt, A., Pennacchiotti, M.: Fate: A framenet-annotated corpus for textual entailment. In: Proceedings of the Sixth conference on International Language Resources and Evaluation, Marrakesh, ELRA (2008)
12. McConville, M., Dzikovska, M.: Using inheritance and coreness sets to improve a verb lexicon harvested from FrameNet. In: Proceedings of the Second Linguistic Annotation Workshop, ELRA (2008) 33–40
13. Matsubayashi, Y., Okazaki, N., Tsujii, J.: A Comparative Study on Generalization of Semantic Roles in FrameNet. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore, ACL (2009) 19–27
14. Fürstenau, H., Lapata, M.: Semi-Supervised Semantic Role Labeling. In: Proceedings of the 12th Conference of the European Chapter of the ACL, Athens, ACL (2009) 220–228
15. Fürstenau, H., Lapata, M.: Graph Alignment for Semi-Supervised Semantic Role Labeling. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, ACL (2009) 11–20
16. Pennacchiotti, M., De Cao, D., Basili, R., Croce, D., Roth, M.: Automatic induction of framenet lexical units. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Honolulu, ACL (2008) 457–465
17. Erk, K., Pado, S.: Shalmaneser - a flexible toolbox for semantic role assignment. In: Proceedings of the Fifth conference on International Language Resources and Evaluation, Genoa, ELRA (2006)
18. Johansson, R., Nugues, P.: Syntactic representations considered for frame-semantic analysis. In: Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories, Bergen (2007)
19. Das, D., Schneider, N., Chen, D., Smith, N.: Probabilistic frame-semantic parsing. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, Los Angeles, ACL (2010) 948–956

JOSEF RUPPENHOFER
SAARLAND UNIVERSITY
COMPUTATIONAL LINGUISTICS AND PHONETICS
SAARBRÜCKEN,
GERMANY
E-MAIL: <JOSEFR@COLI.UNI-SAARLAND.DE>

MANFRED PINKAL
SAARLAND UNIVERSITY
COMPUTATIONAL LINGUISTICS AND PHONETICS
SAARBRÜCKEN,
GERMANY
E-MAIL: <PINKAL@COLI.UNI-SAARLAND.DE>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 41–51
RECEIVED 01/11/10 ACCEPTED 26/11/10 FINAL 08/02/11

Integrated Lexicographic Platform for the Russian Language

ALEŠ HORÁK,[1] MARIA KHOKHLOVA,[2] ADAM RAMBOUSEK,[1] AND VICTOR ZAKHAROV[2]

[1] *Masaryk University, Czech Republic*
[2] *Saint Petersburg State University, Russian Federation*

ABSTRACT

This paper deals with the first phase of building an integrated lexicographic platform for the Russian language which can be used for various linguistic purposes. The aim of this paper is to present the main ideas of the lexicographic platform and linked projects (digitization of explanatory dictionaries of Russian, design and implementation of a database of citations) and to describe the current state of the data sources and lexicographic tools for Russian. This project is realized in cooperation between the Philological Faculty, St. Petersburg State University, Institute for Linguistic Studies, Russian Academy of Sciences, and the Faculty of Informatics, Masaryk University, Brno, Czech Republic. The ultimate aim is to provide new lexicographic software tools for developing explanatory dictionaries of the Russian language.

1 INTRODUCTION

The information society has become very quickly a computerized one. Constantly, new technologies come to new spheres of human activity. The arrival of corpus linguistics and corpora have become a relevant point in this respect. The corpora stimulated a considerable progress that has been gained in the field of automatization of lexicographic work. This has its own reason. There is no integrated software that enables to work both with traditional dictionaries and new electronic sources of lexical data.

The first explanatory dictionaries of Russian date as back as to the beginning of the 19th century. Among dictionaries of contemporary Russian we can name Ushakov's Dictionary (the Explanatory dictionary of the Russian Language, 1935-1940, the 2. revised edition 1947-1948) [1], Ozhegov's Dictionary (the Dictionary of the Russian Language, the first edition was published in 1949) [2], the Dictionary of the Contemporary Russian Language in 17 volumes (also known as BAS – "Bol'shoj akademicheskij slovar' russkogo jazyka", 1950-1965) [3], the Dictionary of the Russian Language in 4 volumes (also known as MAS – "Malyj slovar' russkogo jazyka", 1957-1961, the 2. revised edition 1981-1984) [4], the Complex Normative Dictionary of the Modern Russian Language ("Komplexsnyj normativnyj slovar' sovremennogo russkogo jazyka") [5], and the Big Academic Dictionary of the Russian Language in 25 volumes (also known as the new BAS – "Bol'shoj akademicheskij slovar' russkogo jazyka", since 2004) [6].

The intention is to collect resources of these dictionaries within one framework. All these data will be converted into a well-structured format (e.g., XML format) and concentrated in a unified database. Such a database will be prepared for all kinds of linguistic research.

The idea has been existing for several years and was inspired by several similar projects abroad, as the Celex database [7], and the Czech lexical database [8, 9].

2 DEB PLATFORM

The basis of the new project implementation is formed by the DEB II dictionary writing systems platform developed at the Natural Language Processing Centre, Faculty of Informatics, Masaryk University.

The DEB II system (Dictionary Editor and Browser, `http://deb.fi.muni.cz/`) is an open-source software platform designed for fast development of applications for viewing, creating, editing and authoring of electronic and printed dictionaries. The platform is based on the approach of the client-server architecture (see the DEB II platform schema in Figure 1). Most of the functionality is provided by the server side, and the client side offers (computationally simple) graphical interfaces to users. The client applications communicate with the server using the standard web HTTP protocol.

The server part is built from small reusable parts, called servlets, which allow a modular composition of all services. Each servlet pro-

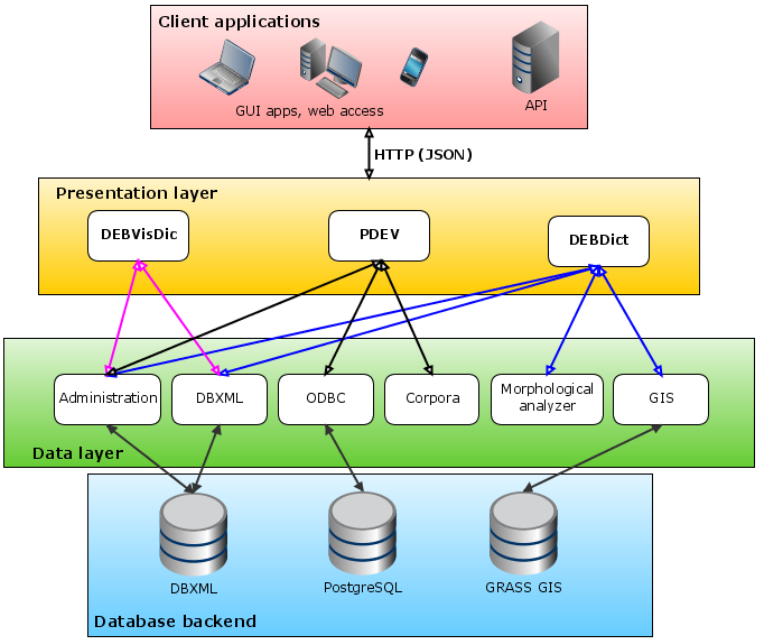


Fig. 1. The DEB II platform schema

vides different functionality such as database access, dictionary search, morphological analysis or a connection to various corpora.

The overall design of the DEB II platform focuses on modularity. The data stored in a DEB II server can employ any kind of structural database and combine the results in answers to user queries without the need to use specific query languages for each data source. The main data storage is currently provided by the Oracle Berkeley DB XML [10]. However, it is possible to switch to another database backend easily, without any changes to the client parts of the applications.

The main assets of the DEB II development platform can be characterized by the following points:

– All the data are stored on the server and a considerable part of the functionality is also implemented on the server, while the client application can be very lightweight.

– Very good tools for team cooperation; data modifications are immediately seen by all the users. The server also provides authentication and authorization tools.
– Server may offer different interfaces using the same data structure. These interfaces can be reused by many client applications.
– Homogeneity of the data structure and presentation. If an administrator commits a change in the data presentation, this change will automatically appear in every instance of the client software.
– Integration with external applications, for example geographic information system or corpus query tools.

The DEB II platform versatility is apparent in more than ten projects based on the platform, ranging from dictionary viewers to complex ontology editors. In the following sections, we provide overview information on the main DEB II applications.

2.1 *DEBDict*

General dictionary browser, used by more than 700 users to access six electronic dictionaries of Czech and other lexical resources. Thanks to the features of the DEB II platform, DEBDict can check user's access rights and thus provide access to selected dictionaries intended for a specific group of people. For example, if the dictionary copyright does not allow public distribution, the access to the dictionary data may be limited to members of a research team.

2.2 *DEBVisDic*

The specific task of preparation of lexical semantic networks with the structure of the Princeton WordNet [11] requires special tools. During the Balkanet project [12], a wordnet browser and editor VisDic was developed by FI MU and it was used for building several national wordnets. Since 2005, it was replaced by DEBVisDic, a new system based on the DEB II development platform.

The DEBVisDic client application is split to the core module and individual modules for each wordnet. This way, it is possible to define different data structure, workflow, or include data from external sources separately for each (national) wordnet. For example, verbs in the Czech wordnet are connected to the verb valency lexicon VerbaLex [13]. The DEBVisDic server part provides an Application Programming Interface (API) usable by external applications or web services.

DEBVisDic was used as a basis for several multilingual projects – the Global Wordnet Grid [14], aiming to gather freely available wordnets of many languages, Cornetto [15], Dutch lexical semantic database, and KYOTO [16], European project building a multilingual knowledge extraction system.

2.3 *PRALED*

The Prague Lexical Database application (called PRALED) is developed in close cooperation with the linguists of the Institute of Czech Language (ICL), Czech Academy of Sciences. The application is used to build the new complex Czech Lexical Database, combining digitized dictionaries with graphical presentations of original (often hand-written) excerpt cards, several text and spoken corpora, morphological analyzer and other resources.

The PRALED users are divided into two groups: the ICL researchers are able to view and create entries, whereas others (usually reviewers) can only view finished entries. Currently, 25 linguists are using the application, each of them is creating over 200 entries per day. To add word usage evidence from the corpora, PRALED is connected with the Czech National Corpus [17].

2.4 *Art Glossaries*

In a joint project with the Faculty of Fine Arts, Brno University of Technology DEB II platform was employed as a base for the multi-lingual glossary of fine arts terms. Textual information was enhanced with the multimedia files – pronunciation recording, graphic samples, or explanatory animations. The glossary contains approximately 2000 entries and is utilized by the students as a helpful educational resource (currently on-line and textbook publishing is being considered).

Following the success of the fine arts glossary, the tool is now being enhanced for a new project in coopeartion with the Theatre Faculty, Janáček Academy of Music and Performing Arts, Brno.

2.5 *Family Names in UK*

One of the recent projects is the application to compile a database of English surnames developed for the University of the West of England.

The database will contain the meanings and origins of up to 150 000 UK surnames and will be made publicly available on-line.

The application is linked to several related resources to provide as many information as possible – surnames' frequency and location, dictionaries, genealogical sources.

3 ELECTRONIC DICTIONARIES OF RUSSIAN

Nowadays many dictionaries of the Russian language (including explanatory ones) exist in an electronic form. But usually these are scanned texts in either graphical or text formats. Lack of structuring makes it difficult to search in them and combine them effectively with other language resources.

Several Russian explanatory dictionaries are available on-line (through Feb-web: Fundamental Electronic Library [3]): Ushakov's Dictionary, the Dictionary of the Russian Language in 4 volumes, and the Dictionary of the Russian Language of the 18th century [18].

There is an option to look up only in one dictionary at the same time and browse in it but not to use it as a database. Because entries of different dictionaries have various structures that makes it hard to work with the data.

This raises the question of one integrated structure of Russian explanatory dictionaries and their conversion to this structure. Moreover, this also leads to the question of developing one tool that could be used both as browser and editor.

For the first stage of the project, we have chosen two dictionaries of Russian. They are the "Complex Normative Dictionary of the Modern Russian Language" ("Komplexsnyj normativnyj slovar' sovremennogo russkogo yazyka") [5] and the above mentioned Dictionary of the Russian Language in 4 volumes [4].

The "Complex Normative Dictionary of the Modern Russian Language" as well as the "Normative Explanatory Dictionary of the Live Russian Language" are being compiled at the Laboratory of Computational Lexicography of the Philological Faculty of St. Petersburg State University (Russia) under the guidance of Prof. G.N. Sklyarevskaya. It is intended for users to provide them with information on correct word usage of latest and newest terms and concepts of modern Russia. The

[3] `http://feb-web.ru`

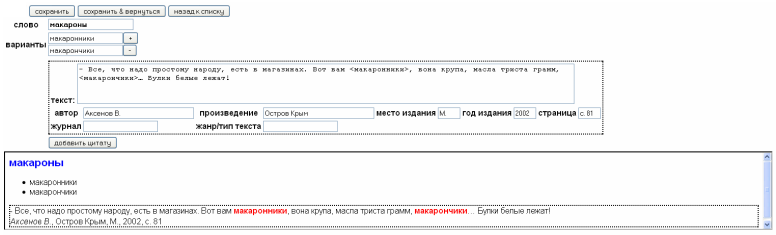


Fig. 2. Example entry (*makarony/pasta*) in the DEB II quotations editor tool

dictionary includes active vocabulary whose selection was based on expert decisions about semantic, grammatical, orthoepic or other features difficult for the language users. The usage of these words has to be normalized. The data is being actively revised and supplemented on the basis of corpus examples, Internet data, various terminological or explanatory dictionaries, and linguistic studies. Dictionary word list is compiled on the data of the Fund of Modern Russian (approximately 17 million tokens).

The DEBDict server was installed at the Institute for Linguistic Studies and the two dictionaries were imported. Although each of them is represented with different XML structure, users are presented with the data in a unified form. Thus lexicographers have obtained access to a valuable research resource, forming the first basic part of the new lexicographic platform.

4 Quotation Database

The Large Card File (LCF) of the Institute for Linguistic Studies of the Russian Academy of Sciences, containing about 8 million of systematized cards with citations, allows for various types of lexicographical and philological research [19]. Its stock was used by lexicographers while compiling a great number of dictionaries and grammars of Russian. Many researchers both from Russia and from abroad use the Large Card File in their investigations on various topics.

The Large Card File was established in the 19th century and currently consists of two parts. One comprises about 5.5 million cards (collected from 1886 to 1968), while the other contains more than 2.5 million cards (collected from 1968 to 1994).

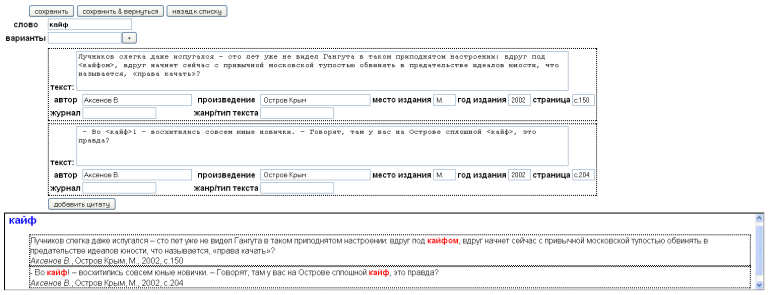


Fig. 3. Example entry (*kajf/pleasure*) in the DEB II quotations editor tool, with multiple quotations

At its present form the card file is not representative enough. This can be accounted for by both its inherent defects (as during the Soviet time a number of authors and works could not be included due to ideological reasons), and by lack of finance – as a consequence for the last 15 years very small amount of new entries have been added to it. It is obvious that only cutting edge information technologies, i.e. electronic libraries, text corpora, programs for lexicographical tasks, can take care of current lexicography needs. Thus, further development and expansion of the LCF should be done electronically.

The final aim is to digitize the content of all the cards in graphical form and build an electronic index of the quotations to help with searching for the headwords, authors etc.

However, digitization of the whole card file is expensive and time-consuming. In the first stage, the newly acquired quotations will be entered directly in the electronic database. During the testing stage, software tools can be enhanced to meet the needs of the users and project. All cards will be digitized, if the evaluation of testing stage is successful and additional funding allows it.

The quotation database is implemented on the DEB II platform. The user interface is formed by a web application, thus the users do not need to install any special extensions. See the interface example in Figures 2 and 3.

During the development of the database, the DEB II platform was also enhanced with new features needed for the Russian lexicographic tools. A new method of user interface localization was implemented that allows easy updates of the texts in any language and any character set. All the

interface texts are stored in a XSLT file which is transformed into several formats and included in the set of XSLT templates, JavaScript files and internal templates.

The database is connected with the Russian National Corpus and the DEBDict service with Russian electronic dictionaries, taking a step further to the desired lexicographic platform. Linguists do not need to run several applications, they can work with several resources within one tool.

Before the development of the database, new quotations were tentatively collected in text files, these were converted to the XML format and imported into the database. Currently, the database contains over 2200 quotations for 2000 words.

5 Conclusion

In this paper, we have presented the results of the first phase of the development of new lexicographic platform for the Russian language. The final aim of this project is to fill in the gap in providing complex software tools based on standard technologies which offer the unified presentation of current Russian lexicographic resources.

The developed platform is based on the DEB II framework, which has currently been used in several international projects for preparing new specialized applications for presentation and editing of lexicographic resources of various kinds and purposes. We believe that the resulting system will enhance the Russian lexicographic work by processing the current rich set of resources with specialized language technologies.

Acknowledgements

The work is supported by the grant of the Russian Foundation for Basic Research No. 10-07-00563A and by the grant of the Russian Foundation for Humanities No. 10-04-12135B.

This work has been partly supported by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009 and by the Czech Science Foundation under the projects P401/10/0792.

References

1. Ushakov, D.N., ed.: Tolkovyj slovar' russkogo jazyka v 4 tomakh. (1935-1940)

2. Ozhegov, S.I.: Slovar' russkogo jazyka, Moscow (1949) Ed. by S. P. Obnorskij.
3. Babkin, A.M., Barkhudarov, S.G., Philin, P.P., eds.: Slovar' sovremennogo russkogo literaturnogo jazyka v 17 tomakh, Moscow, Leningrad (1950-1965) Widely used abbreviation BAS.
4. Jevgen'jeva, A.P., ed.: Slovar' russkogo jazyka v 4 tomakh, Moscow (1957-1961) Widely used abbreviation MAS.
5. : Komplexsnyj normativnyj slovar' sovremennogo russkogo jazyka (2010)
6. Gerd, A.S., ed.: Bol'shoj Akademicheskyj slovar' russkogo jazyka Rossijskoj akademii nauk, St. Petersburg (since 2004) volume 1-14.
7. Celex: Celex lexical database (2004) `http://www.ldc.upenn.edu/Catalog/readme_files/celex.readme.html`.
8. Klímová, J., Oliva, K., Pala, K.: Czech lexical database – first stage. In: Short Proceedings of Complex Conference 2005, Budapest, Hungary (April 2005)
9. Rangelova, A., Králík, J.: Wider Framework of the Research Plan Creation of a Lexical Database of the Czech Language of the Beginning of the 21st Century. In: Proceedings of the Computer Treatment of Slavic and East European Languages 2007, Bratislava, Slovakia (2007) 209–217
10. Chaudhri, A.B., Rashid, A., Zicari, R., eds.: XML Data Management: Native XML and XML-Enabled Database Systems. Addison Wesley Professional (2003)
11. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. MIT Press (1998)
12. Horák, A., Smrž, P.: VisDic – wordnet browsing and editing tool. In: Proceedings of the Second International WordNet Conference – GWC 2004, Brno, Czech Republic (2003) 136–141 `http://nlp.fi.muni.cz/projekty/visdic/`.
13. Hlaváčková, D., Horák, A.: VerbaLex – New Comprehensive Lexicon of Verb Valencies for Czech. In: Proceedings of the Slovko Conference, Bratislava, Slovakia (2005)
14. Horák, A., Pala, K., Rambousek, A.: The Global WordNet Grid Software Design. In: Proceedings of the Fourth Global WordNet Conference, Szegéd, Hungary, University of Szegéd (2008)
15. Horák, A., Vossen, P., Rambousek, A.: A Distributed Database System for Developing Ontological and Lexical Resources in Harmony. In: Lecture Notes in Computer Science: Computational Linguistics and Intelligent Text Processing, Haifa, Israel, Springer-Verlag (2008) 1–15
16. Vossen, P.: KYOTO Project (ICT-211423), Knowledge Yielding Ontologies for Transition-based Organization (2008) `http://www.kyoto-project.eu/`.
17. ICNC: Czech National Corpus - SYN2000 (2000) Accessible at WWW: `http://www.korpus.cz`.
18. Sorokin, J.S., ed.: Slovar' russkogo jazyka XVIII veka, Leningrad-St. Petersburg (since 1984)

19. Rogozhnikova, R.P.: Sokrovishchnitsa russkogo slova. Istoriia bolshoi slovarnoi kartoteki Instituta lingvisticheskikh issledovanii RAN, Saint-Petersburg, Russian Federation (2003)

ALEŠ HORÁK
FACULTY OF INFORMATICS,
MASARYK UNIVERSITY,
BOTANICKÁ 68A, 602 00 BRNO, CZECH REPUBLIC
E-MAIL: <HALES@FI.MUNI.CZ>

MARIA KHOKHLOVA
PHILOLOGICAL FACULTY,
SAINT PETERSBURG STATE UNIVERSITY,
UNIVERSITY EMB. 11, 199034 ST. PETERSBURG
RUSSIAN FEDERATION
E-MAIL: <KHOKHLOVA.MARIE@GMAIL.COM>

ADAM RAMBOUSEK
FACULTY OF INFORMATICS,
MASARYK UNIVERSITY,
BOTANICKÁ 68A, 602 00 BRNO, CZECH REPUBLIC
E-MAIL: <XRAMBOUS@FI.MUNI.CZ>

VICTOR ZAKHAROV
PHILOLOGICAL FACULTY,
SAINT PETERSBURG STATE UNIVERSITY,
UNIVERSITY EMB. 11, 199034 ST. PETERSBURG
RUSSIAN FEDERATION
E-MAIL: <VZ1311@YANDEX.RU>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 53–72
RECEIVED 05/11/10 ACCEPTED 02/12/10 FINAL 09/02/11

A Constraint Based Hybrid Dependency Parser for Telugu

SRUTHILAYA REDDY KESIDI, PRUDHVI KOSARAJU,
MEHER VIJAY, AND SAMAR HUSAIN

IIIT-Hyderabad, India

ABSTRACT

In this paper we present a two stage constraint based approach to Telugu dependency parsing. We define the two stage and show how different Telugu constructions are parsed at appropriate stages. The division leads to selective identification and resolution of specific dependency relations at the two stages. We then discuss the ranking strategy that makes use of the S-constraints to gives us the best parse. A detailed error analysis of the output of core parser and the ranker helps us to flesh out the current issues and future directions.

KEY WORDS: Telugu dependency parser, constraint based parsing, parse ranking, hybrid approach, morphologically rich free word order parsing

1 INTRODUCTION

There has been a recent surge in addressing parsing for morphologically rich free word order languages such as Czech, Turkish, Hindi, etc. These languages pose various challenges for the task of parsing mainly because the syntactic cues necessary to identify various relations are complex and distributed [14, 1, 10, 11, 13, 12, 6, 9, 5]. Data driven parser performance [7] show that many of these complex phenomena are not being captured by the present parsers.

Constraint based parsers have been known to have the power to account for difficult constructions and are well suited for handling complex phenomena. Some of the recent constraint based systems have been [20, 24, 36, 37, 31, 28, 23, 17].

In this paper we present a two stage constraint based approach to Telugu dependency parsing based on the parser proposed by [3, 4, 35]. We begin by quickly summarizing the parser design and follow it by describing how different Telugu constructions are parsed at appropriate stages. We will see that this division leads to selective identification and resolution of specific dependency relations at the two stages. We then discuss the ranking strategy that makes use of the S-constraints to gives us the best parse. A detailed error analysis of the output of core parser and the ranker helps us to flesh out the current issues and future directions.

This paper is arranged as follows: Section 2 gives a quick overview of the two-stage constraint based hybrid parser (CBHP); Section 3 explains in details how we handle different Telugu constructs in two stages; Section 4 describes the results of the core parser and makes some observations on these results. In Section 5 we describe the ranking strategies, followed with results in Section 6. We conclude the paper with future directions in Section 7.

2 OVERVIEW OF THE TWO-STAGE CONSTRAINT BASED HYBRID PARSER (CBHP)

CBHP [3, 4] adopts a hybrid approach to dependency parsing. A constraint based approach is supported by a controlled statistical strategy to achieve high performance and robustness. The overall task of dependency parsing is attacked using modularity, wherein specific tasks are broken down into smaller linguistically motivated sub-tasks. Figure 1 shows the overall design of CBHP. Below we quickly summarize CBHP:

(1) *Two stages during the sentence analysis phase:* The parser tries to analyze the given input sentence, which has already been tagged and chunked,[1] in two stages; it first tries to extract intra-clausal[2]

[1] POS and chunk tagging scheme for Indian Languages is discussed in [38].

[2] A clause is a group of word such that the group contains a single finite verb and its dependents. We must note here that these dependents cannot

dependency relations. In the second stage it then tries to handle more complex inter-clausal relations such as those involved in constructions of coordination and subordination between clauses.

(2*) H-constraints:* Both 1st and 2nd stage use linguistically motivated constraints. These H- constraints (Hard constraints) reflect that aspect of the grammar which in general cannot be violated. H-constraints mainly comprised of structural and lexical knowledge of the language.

(3) *S-constraints and Prioritization for parse selection*: The sentence analysis layer (cf. figure 1) can potentially produce more than one parse. These parses are then ranked by a prioritization component using S-constraints(Soft constraints). S-constraints are the constraints that are used in the language as preferences. Unlike the H-constraints that are derived from a knowledge base and are used within the core parser during derivation, S-constraints have weights assigned to them and are used exclusively during prioritization. These weights are automatically learnt using a manually annotated dependency treebank.

3 CBHP FOR TELUGU

[3], [4] have proposed CBHP for Indian languages. It has however been tested only for Hindi. We use the same machinery that was used by them to handle Telugu. We first describe the different types of relations the parser currently handles in the two stages, following which we briefly mention H-constraints for Telugu CBHP.

3.1 RELATIONS HANDLED IN STAGE 1

In stage 1 we handle mainly intra-clausal relations. These relations represent:

---

themselves be finite verbs. Also, subordinating conjunctions and finite verb coordinating conjunctions are also not treated as part of a clause. And therefore, a sentence such as '*John said that He will come late*' has 3 units; (1) *John said*, (2) *that*, and (3) *He will come late*. Similarly, '*John ate his food and he went shopping*' has 3 units; (1) *John ate his food*, (2) *and*, and (3) *he went shopping*.

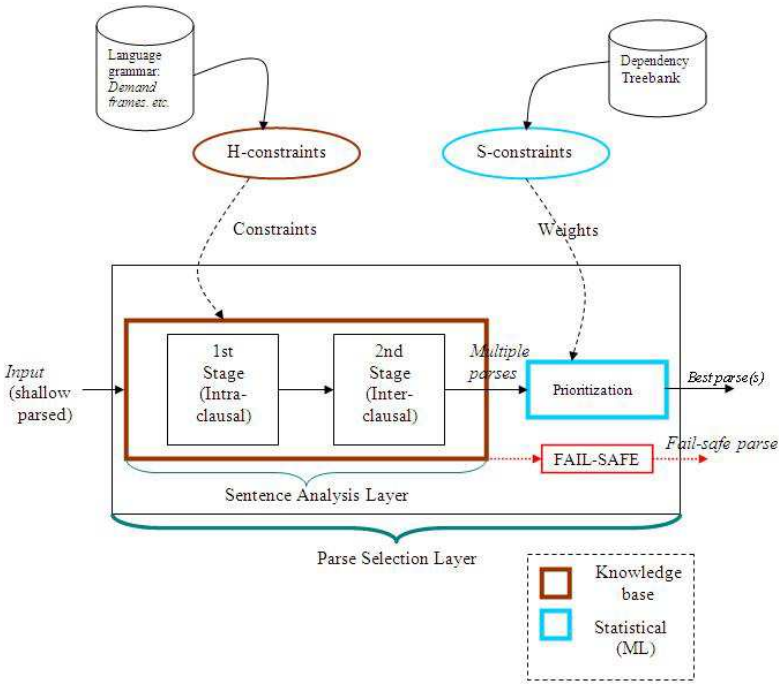


Fig. 1. Design of CBHP

i. Argument structure of the finite verb
ii. Argument structure of the non-finite verb
iii. Adjuncts of finite and the non-finite verb
iv. Noun modifications
v. Adjectival modifications

Using example (1) below we describe how the parser handles a simple declarative sentence.

(1) *wulasi golilu mAnesiMxi*
'Tulasi' 'tablets' 'stopped using'
'Tulasi stopped using the tablets'

CBHP begins by constructing the constraint graph (CG) for the above sentence. A constraint graph shows all the potential arcs that can exist between the heads and their corresponding dependents. This

information is obtained from the demand frame of the head. According to the frame the verb can take k1[3] and k2 as mandatory children with null postpositions. Figure 2 shows that the demand frame for *mAnesiMxi* is obtained from basic demand frame of root verb *mAnu* and the null frame of suffix *–esiMxi* (which represent the tense, aspect and modality (TAM)). The basic demand frame for a verb or a class of verbs specifies the suffix, category, etc. permitted for the allowed relations for a verb when the verb has the basic TAM label. In Figure 3(a) we show the constraint graph that is constructed using the demand frame. The final parses are obtained by converting the CG into integer programming equations and using bi-partite graph matching [36, 37, 4]. Figure 3(b) shows the final parses obtained from the CG. Notice that we get two parses. This indicates the ambiguity in the sentence if only the limited knowledge base is considered. Stated another way, H-constraints (in the form of demand frames) are insufficient to restrict multiple analysis of a given sentence and that more knowledge (semantics, other preferences, etc.) is required to curtail the ambiguities. We will see in Section 5 how we can obtain the best parse from these multiple parses. Notice also the presence of the _ROOT_ node. By introducing _ROOT_ we are able to attach all unprocessed nodes to it, ensuring that the output we get after each stage is a tree.

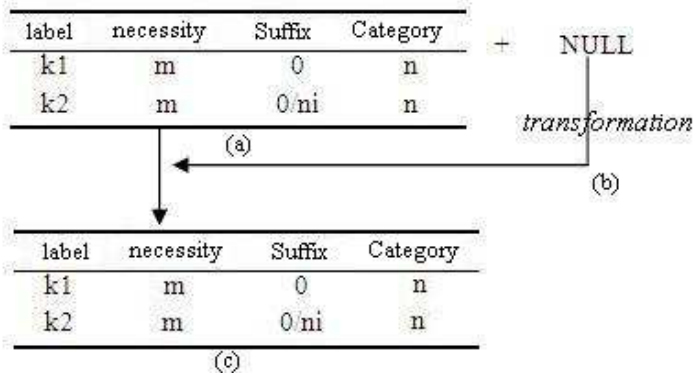
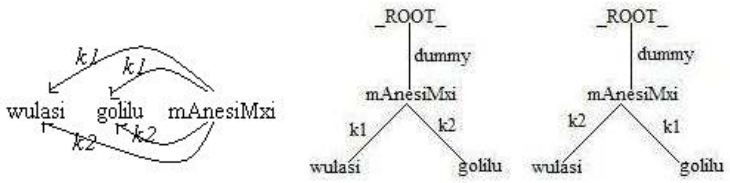


Fig. 2. Final demand frame for *mAnesiMxi* shown in (c) is obtained from the basic demand frame of *manu* (a) and the transformation (b) which is NULL here.

[3]k1 can be roughly translated to 'agent', k2 can be roughly translated as 'theme'. CBHP follows the syntactic-semantic dependency labels proposed in [3].



(a) **(b)**

Fig. 3. (a) Constraint graph for (1). (b) Final parses obtained from CG

Example (2) is a slightly complex construction containing a non-finite verb. Such sentences are also handled in the 1st stage.

(2) *wulasi rogaM waggiMxani golilu mAnesiMxi.*
'Tulasi' 'disease' 'having reduced-so' 'tablets' 'stopped using'.
'As the disease reduced, Tulasi stopped using the tablets'

We have already seen through (1) how a sentence with a finite verb like *mAnesiMxi* can be parsed. In (2) in addition to a finite verb we have a non-finite verb *waggiMxani*. Like last time we first get the basic demand frame of the non-finite verb *waggiMxani*. Basic demand frames always represent the argument structure of a verb with its default TAM (present, indefinite). When a new TAM appears with the verb, we transform the original frame to get the final frame that is accessed during the construction of the CG. Figure 4 shows this process clearly. Figure 5 shows the most appropriate final parse.

3.2 RELATIONS HANDLED IN STAGE 2

Stage 2 handles more complex inter-clausal relations such as those involved in constructions of coordination and subordination between clauses. Stage 2 handles the following relations:

i. Clausal coordination
ii. Clausal subordination
iii. Non-clausal coordination
iv. Clausal complement

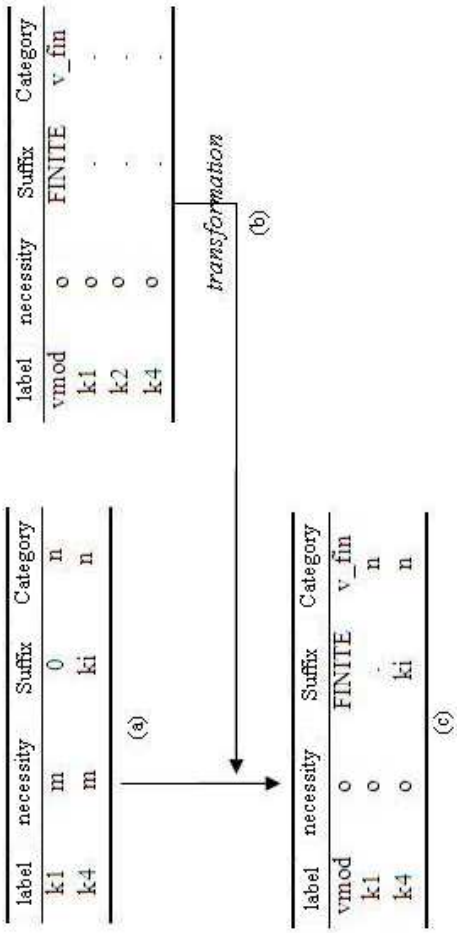


Fig. 4. Basic Demand frame of *waggu* (a) is transformed by the transformation frame *A_ani* (b) giving us the final frame for *waggiMxani* shown in (c).

2nd stage functions similar to the 1st stage, except that in the 2nd stage the CG has very few nodes. This is because the 1st stage parse subtrees are mostly not modified in the 2nd stage and only those nodes that are relevant for 2nd stage relations are considered. Take (3) as a case in point.

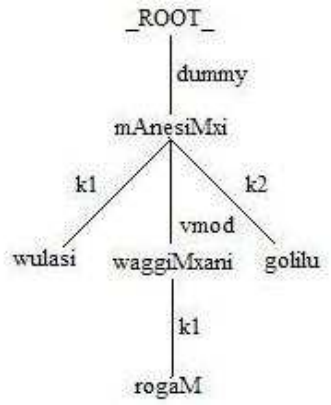


Fig. 5. Appropriate final parse

(3) *wulasi golilu mAnesiMxi mariyu paniki velYliMxi*
'Tulasi' 'tablets' 'stopped using' 'and' 'work' 'went'
'Tulasi stopped using tablets and went to work'

Figure 7(a) shows the first stage parse for this sentence. We can see that the analysis of the two clauses is complete. The root of these subtrees and the conjunct *mariyu* are seen attached to the _ROOT_. Figure 6 shows the 2nd stage CG for (3). Notice that only the two finite verbs and the conjunct are seen here. The relations identified in the 1st stage remain untouched. Figure 7(b) also shows the final parse for this sentence; notice that the outputs of the two stages are combined to get the final parse. Merging the 1st stage and 2nd stage is not problematic as the two stages are mutually exclusive.

We must note here that for few cases such as (4) below, the 1st stage output is revised. We follow the same principles as described in [35]. This mainly concerns case dropping in nominal coordinations. Example 5 shows an example of clausal complement where the verb *ceVppiMxi* takes the clause headed by a complementizer *ani*, the complementizer in turn takes a clause headed by *mAnesiMxi.* Figure 8(a) and 8(b) shows the 1st and 2nd stage output of (5) respectively. The 2nd stage CG can be seen in Figure 9.

(4) *wulasi mariyu rama golilu mAnesAru.*
'Tulasi 'and' 'Rama' 'tablets' 'stopped using'
'Tulasi and Rama stopped using tablets.'

(5) *wulasi golilu mAnesiMxi ani rama ceVppiMxi*
'tulasi' 'tablets' 'stopped using' '' 'rama' 'told'
'Rama told that Tulasi stopped using tablets.'

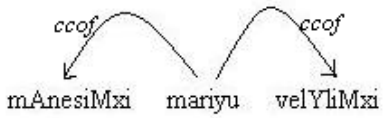
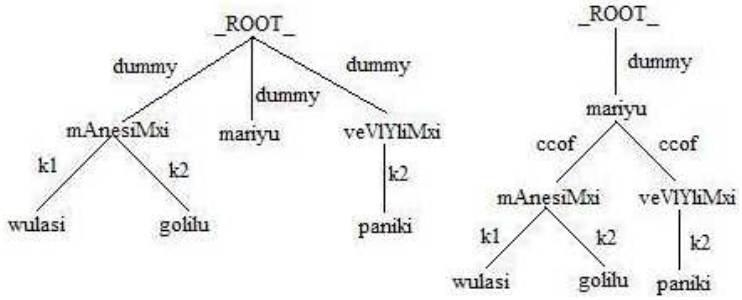


Fig. 6. 2nd stage constraint graph for (3)

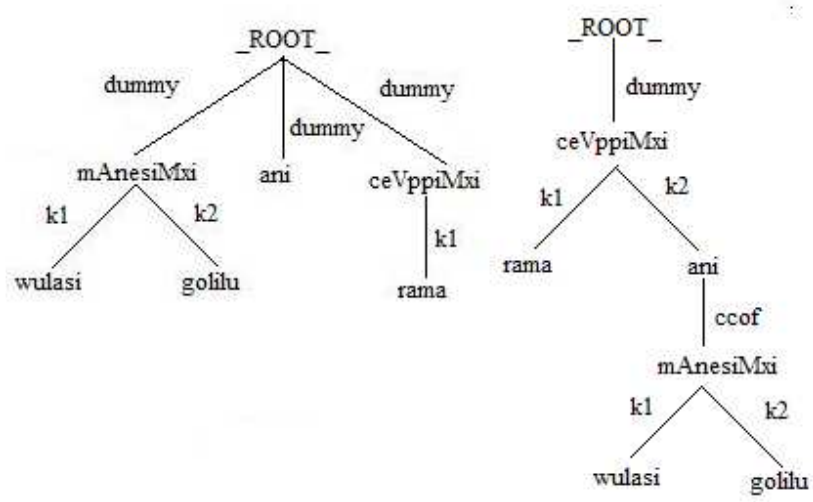


(a) **(b)**

Fig. 7. (a) Stage 1 and (b) Stage 2 parse output for (3)

3.3 H-CONSTRAINTS

As mentioned earlier H-constraints in CBHP mainly comprises of the lexical constraints. This as we saw in section 3.1 corresponds to the basic demand frame and the transformation frame. For Telugu CBHP we manually prepared around 460 verb frames and 95 transformation frames. The transformation frames handles various alternations that are brought in by non-finite and passive TAMs. High frequency verbs and tense, aspect and modality markers were extracted from the training data to prepare the frames. Similarly, other heads such as conjuncts were also extracted. Preparation of these frames took around 30 days.



(a) (b)

Fig 8. Stage 1 and Stage 2 outputs for sentence (5)

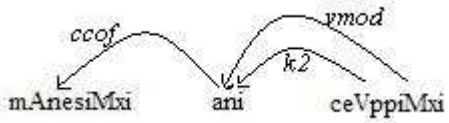


Fig 9. 2nd stage CG for sentence (5)

4 RESULTS AND OBSERVATIONS

In this section we describe the data that was used for evaluation. We then give the oracle result of the core parser on this data, following which we discuss the results and the error analysis. The oracle parse for a sentence is the best available parse amongst all the parses of that sentence. It is obtained by selecting the parse closest to the gold parse. The oracle accuracy gives the upper-bound of the parser accuracy and gives some idea about its coverage.

4.1 DATA

All the results in this paper are reported for Telugu. We use the Telugu data set that was released as part of the ICON Tools Contest 2010 [8]. The training data had 1300 sentences, development and test set had 150 and 150 sentences respectively. Since the released data is a preliminary version of the treebank it had few errors. Certain relations related to experiencer verbs and verbs of movement have been corrected to report the results. The current parser does not handle ellipses and therefore all the sentences with NULL nodes have been removed to report the results. This data has 1119 training, 133 development and 127 test sentences.

Table 1. Overall parser oracle accuracy

	LAS	UAS	LS
Development	68.06	84.41	70.34
Testing	65.33	84.14	66.60

4.2 RESULTS

Table 1 below shows the oracle accuracies of the parser for the development and testing data. We see that the UAS (unlabeled attachment score) for both test and development is very good; the accuracies for LAS (labeled attachment score) and LS (labeled score) however are not. In Table 2 we show the breakup of the results into intra-clausal and inter-clausal relations. We see that on an average the inter-clausal relations are being identified successfully, and the low LAS of Table 1 can be attributed mainly to the intra-clausal relations.

Table 2. Intra-clausal and Inter-clausal relation accuracy

		LAS	UAS	LS
Development	**Intra-clausal**	59.83	82.82	63.15
	Inter-clausal	85.89	87.73	85.89
Test	**Intra-clausal**	57.92	85.11	59.87
	Inter-clausal	79.63	82.09	79.63

Further analysis of the results showed why the oracle LAS is not very high:

1. *Coverage of H-constraints*: As mentioned earlier we are currently using around 460 frames in the parser. Close to 30% of all the verbs in the test and development data were unseen. The parser uses a default strategy for unseen verbs; not surprisingly, this does not always work well. Similar observation has been reported in the literature for all the parsing approaches in general [22].

2. *Unhandled Relations*: There are still some relations that the parser doesn't handle. Complex predicate is one such case. Automatic identification of such predicates is a challenging task as most diagnostics proposed in the literature are behavioral [39, 32]. There has been some work in automatically identifying complex predicates for Hindi [25, 30]; we need to try and see if these methods can help us too. Ellipses is another phenomena that the parser doesn't handle. In Telugu, sometimes even the main arguments of the verb might go missing and in such cases the parser might assign this relation to some other word with the same property.

3. *Morphological errors and ambiguous TAMs:* A small portion of errors are caused when the morphological analyzer gets the root form of a verb wrong. In such a case, CBHP will pick incorrect verb frame. Also, in Telugu certain TAM (tense, aspect and modality) labels are ambiguous and will affect transformations.

5 S-CONSTRAINTS AND PRIORITIZATION

It was clear from Sections 2 and 3 that the core parser that uses H-constraints produces multiple parses. S-constraints are those constraints that are used in a language as preferences and hence can be used to rank the multiple parses. These S-constraints can be used for ranking by penalizing a parse for the constraint that it violates and finally choosing the parse that gets least penalized. This strategy is similar to the one used in Optimality Theory [33, 34]. The other way is to use these S-constraints as features, associate weight with them, use them to

score the output parses and select the parse with the best score. We use the latter strategy. The score of a dependency parse tree t=(V, A) in most graph-based parsing system [26] is

$$\text{Score}(t) = \text{Score}(V, A) \in \mathbf{R} \quad \text{(I)}$$

where V and A are the set of vertices and arcs. This score signifies how likely it is that a particular tree is the correct analysis of a sentence S. Many systems assume the above score to factor through the scores of subgraph of t. Thus, the above score becomes

$$\text{Score}(t) = \Sigma_{\alpha \in \alpha t} \lambda_{\alpha} \quad \text{(II)}$$

where α is the subgraph, α_t is the relevant set of subgraph in t and λ is a real valued parameter. If one follows the arc-factored model for scoring a dependency tree [26] like we do, the above score become

$$\text{Score}(t) = \Sigma_{(i, r, j) \in A} \lambda_{(i, r, j)} \quad \text{(III)}$$

In (III) the score is parameterized over the arcs of the dependency tree. Since we are interested in using this scoring function for ranking, our ranking function (R) should therefore select the parse that has the maximum score amongst all the parses (Φ) produced by the core parser.

$$R(\Phi, \lambda) = \text{argmax}_{(t=V,A) \in T} \text{ Score } (t) = \text{argmax}_{(t=V,A) \in T} \Sigma_{(i, r, j) \in A} \lambda_{(i, r, j)} \quad \text{(IV)}$$

Since in our case $\lambda_{(i, r, j)}$ represent probabilities, it is more natural to multiply the arc parameters instead of summing them.

$$R(\Phi, \lambda) = \text{argmax}_{(t=V,A) \in T} \text{ Score } (t) = \text{argmax}_{(t=V,A) \in T} \Pi_{(i, r, j) \in A} \lambda_{(i, r, j)} \quad \text{(V)}$$

For us $\lambda_{(i, r, j)}$ is simply the probability of relation *r* on arc i → j given some S-constraints (Sc). This probability is obtained using the MaxEnt model [40]. So,

$$\lambda_{(i, r, j)} = p(r_{i,j} \mid \text{Sc}) \quad \text{(VI)}$$

If A denotes the set of all dependency labels and B denotes the set of all S-constraints then MaxEnt ensures that p maximizes the entropy

$$H(p) = -\Sigma_{x \in E} p(x) \log p(x) \quad \text{(VII)}$$

where $x = (a,b)$, $a \in A$, $b \in B$ and $E = A \times B$. Note that, since we are not parsing but prioritizing, unlike the arc-factored model where the feature function associated with the arc parameter consists only of the

features associated with that specific arc, our features can have wider context. Figure 10 shows the context over which various S-constraints can be specified to create the MaxEnt model. Some of the S-constraints that have been tried out are: (1) *Order of the arguments*, (2) *Relative position of arguments with respect to the verb*, (3) *Agreement*, (4) *General graph properties*.

These S-constraints get reflected as features that are used in MaxEnt. The features for which the model gave the best performance are given below. Note that the actual feature pool was much larger, and some features like that for agreement did not get selected.

(1) Root, POS tag, Chunk tag, suffix of the current node and its parent
(2) Suffix of the grandparent, Conjoined suffix of current node and head
(3) Root, Chunk Tag, Suffix, Morph category of the 1st right sibling
(4) Suffix, Morph category of the 1st left sibling
(5) Dependency relations between the first two, right and left sibling and the head
(6) Dependency relation between the grandparent and head
(7) Dependency relation between the current node and its child
(8) A binary feature to signify if a k1 already exist for this head
(9) A binary feature to signify if a k2 already exist for this head
(10) Distance from a non-finite head

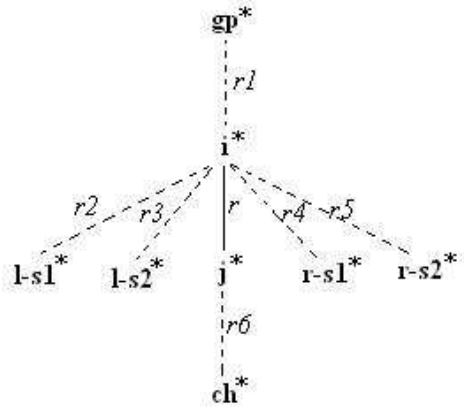


Fig. 10. Context over which S-constraints can be specified. Node *i* is the parent of node j. l–s1 corresponds to 1st left sibling, r–s1 corresponds to 1st right sibling, gp is grandparent of node j, ch is child of node j. r1–r6 are dependency relations

The ranking function shown in (V) can differ based on how one gets the probability of relation on arc i → j. Since we are ranking labeled dependency tree the first way (as shown in VI) is to use the probability of the label *r* in the labeled dependency parse. But we can also use the probability of the label given by the MaxEnt model. Considering this, the third obvious way is to take the weighted average the two method. (VIII) and (IX) show these other two options.

$$\lambda_{(i, r, j)} = p(rm_{i,j} \mid Sc) \quad \text{(VIII)}$$

where, rm is the relation on arc i → j predicted by the model.

$$\lambda_{(i, r, j)} = (p(r_{i,j} \mid Sc) + p(rm_{i,j} \mid Sc)) / 2 \quad \text{(IX)}$$

When the ranker uses (VI) we call it 'Ranking with Parser Relation probability' (Rank-PR), the other two are called 'Ranking with Model Relation probability' (Rank-MR) and 'Ranking with Weighted Relation probability' (Rank-WR).

6 Prioritization Results and Observations

Table 3 shows the result of the MaxEnt model[4] on the development and test data. The features used for training were mentioned in the previous section.

Table 3. Accuracy of the MaxEnt labeler

	Accuracy
Development	76.62
Test	76.78

The result for Rank-PR, Rank-MR, and Rank-WR on both development and testing data is shown in Table 4. It is interesting to note that the best system turns out to be Rank-WR. One should not be surprised with this as this strategy combines the advantage of both the parser labels and the MaxEnt predicted labels. We can see that the best UAS is very close to the oracle UAS. The difference however is wider for LAS.

[4]http://maxent.sourceforge.net/

Table 4. Parser accuracy after Ranking

		LAS	UAS	LA
Development	**Rank-PR**	59.51	81.56	63.12
	Rank-MR	57.03	82.13	61.03
	Rank-WR	59.51	81.94	63.31
Test	**Rank-PR**	58.99	82.45	61.10
	Rank-MR	55.18	81.82	57.72
	Rank-WR	59.83	82.45	61.52

The average number of output parses for each sentence is around 10. It was noticed that the differences between these parses were very minimal and this makes ranking them a non-trivial task. The closeness between parses is quite expected from a constraint based parser whose output parses are only those that do not violate any of the H-constraints. In other words most of the output parses are linguistically very sound. Of course, linguistic soundness is only restricted to morpho-syntax and does not consider any semantics. This is because the H-constraints do not incorporate any semantics in the parser as of now. Considering this, the error analysis doesn't throw up any big surprises. The main reasons why the LAS suffers can be attributed to:

i. *Lack of explicit post-positions or presence of ambiguous one*: Errors because of this, manifest themselves at different places. This can lead to attachment error. Few common cases are finite and non-finite argument sharing, confusion between finite and non-finite argument, adjectival participle, appositions, etc. Also, it was noted that the most frequent errors are for those arguments of the verb, that have no postposition. Consequently, relations such as 'k1', 'k2', 'k7' and 'vmod' have very high confusion. The other major error caused by lack of postposition is the selection of parses with argument ordering errors.

ii. *Multiple parses with the same score*: It is possible that more than one parse finally gets the same score. This is partly caused due the above reason but it also reflects the accuracy of the labeler. As the accuracy of the labeler increases this problem will lessen. Currently, we select only the first parse amongst all the parses with equal score.

7 Conclusion and Future Directions

In this paper we successfully adapted a constraint based hybrid parser for Telugu. We showed that the parser is broad coverage and handles various syntactic phenomena. We motivated the analysis in two stages and showed that a finite clause can be a basis of such a division. The oracle accuracies of the parser on the development and the test data set shows that the parser performs well, however there is lot of room for improvement in LAS. The deficit in LAS, as showed, was due to reasons that can be resolved. Apart from incorporating more H-constraints, handling more constructions, we also plan to try and induce the H-constraints automatically from a treebank. For Hindi and Telugu, this has recently been successfully shown by [21]. Along with the base parser, we also discussed the ranking strategy to get the best parse. We noticed that the best selected parse comes very close to the oracle UAS but lags behind in LAS. The error analysis shows that this is mainly because of lack of any explicit cues in the sentence. One of the things that we plan to do to help improve the final selection is to use an OT style filter [34] to compliment the present ranker. Of course, the ranker also benefits from any improvement in the core parser.

References

1. Ambati, B.R., Husain, S., Nivre, J., Sangal, R.: On the Role of Morphosyntactic Features in Hindi Dependency Parsing. In: NAACL-HLT 2010 workshop on Statistical Parsing of Morphologically Rich Language, Los Angeles, CA. (2010)
2. Begum, R., Husain, S., Dhwaj, A., Sharma, D., Bai, L., Sangal, R.: Dependency annotation scheme for Indian languages. In: IJCNLP08 (2008)
3. Bharati, A., Husain, S., Misra, D., Sangal, R.: Two stage constraint based hybrid approach to free word order language dependency parsing. In: The 11th IWPT09. Paris (2009)
4. Bharati, A., Husain, S., Vijay, M., Deepak, K., Misra, D., Sangal, R.: Constraint Based Hybrid Approach to Parsing Indian Languages. In: The 23rd Pacific Asia Conference on Language, Information and Computation. Hong Kong (2009)
5. Eryigit, G., Nivre, J., Oflazer, K.: Dependency Parsing of Turkish. Computational Linguistics 34(3), 357-389 (2008)
6. Goldberg, Y., Elhadad, M.: Hebrew Dependency Parsing: Initial Results. In: The 11th IWPT09. Paris (2009)

7. Hall, J., Nilsson, J., Nivre, J., Eryigit, G., Megyesi, B., Nilsson M., Saers, M.: Single Malt or Blended? A Study in Multilingual Parser Optimization. In: EMNLP-CoNLL shared task (2007)
8. Husain, S., Mannem, P., Ambati, B., Gadde, P.: The ICON-2010 tools contest on Indian language dependency parsing. In: ICON-2010 tools contest on Indian language dependency parsing. Kharagpur, India (2010) *to appear*
9. Husain, S., Gadde, P., Ambati, B., Sharma, D., Sangal, R.: A modular cascaded approach to complete parsing. In: The COLIPS International Conference on Asian Language Processing (IALP) Singapore (2009)
10. McDonald, R., Nivre, J.: Characterizing the Errors of Data-Driven Dependency Parsing Models. In*:* Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2007)
11. Nivre. J.: Non-Projective Dependency Parsing in Expected Linear Time. In: ACL-IJCNLP (2009)
12. Seddah, D., Candito, M., Crabbé, B.: Cross parser evaluation : a French Treebanks study. In: The 11th IWPT09. Paris (2009)
13. Tsarfaty, R., Sima'an., K.: Relational-Realizational Parsing. In: The 22nd CoLing. Manchester, UK. (2008)
14. Tsarfaty, R., Seddah, D., Goldberg, Y., Kuebler, S., Versley, Y., Candito, M., Foster, J., Rehbein, I., Tounsi, L.: Statistical Parsing of Morphologically Ricj Languages (SPMRL) What, How and Wither. In: NAACL-HLT 2010 workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010), Los Angeles, CA. (2010)
15. Begum, R., Husain, S., Sharma, D., Bai, L.: Developing Verb Frames in Hindi. In: LREC (2008)
16. Collins, M., Koo, T.: Discriminative reranking for natural language parsing. In: CL p.25-70 March05 (2005)
17. Debusmann, R., Duchier, D., Kruijff, G.: Extensible dependency grammar: A new methodology. In: Workshop on Recent Advances in Dependency Grammar, pp. 78–85 (2004)
18. Foth, K. A., Menzel, W.: Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In: COLING-ACL06 (2006)
19. Goldberg, Y., Elhadad, M.: Hebrew Dependency Parsing: Initial Results. In: the 11th IWPT09. (2009)
20. Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A. (eds): Constraint Grammar: A language-independent system for parsing unrestricted text. Mouton de Gruyter. (1995)
21. Kolachina, P., Kolachina, S., Singh, A.K., Naidu, V., Husain, S., Sangal, R., Bharati, A.: Grammar Extraction from Treebanks for Hindi and Telugu. (2009)
22. Manning, C.D., Schütze, H: Foundations of statistical natural language processing. MIT Press, pp. 272, 299 (2002)
23. Martins, A., Smith, N., Xing, E.: Concise Integer Linear Programming Formulations for Dependency Parsing. In: the ACL-IJCNLP09 (2009)

24. Maruyama, H.: Structural disambiguation with constraint propagation. In: ACL:90 (1990)
25. Mukerjee, A., Soni, A.,, Raina, A.M.: Detecting Complex Predicates in Hindi using POS Projection across Parallel Corpora. In: the COLING-ACL Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, Sydney. (2006)
26. Kubler, S., McDonald, R., Nivre, J.: Dependency parsing. Morgan and Claypool. (2009)
28. Schröder, I.: Natural Language Parsing with Graded Constraints. PhD thesis, Hamburg Univ (2002)
29. Shen, L., Sarkar, A., Joshi, A.K.: Using LTAG Based Features in Parse Reranking. In: EMNLP (2003)
30. Sinha, R. M. K.: Mining Complex Predicates In Hindi Using A Parallel Hindi-English Corpus. In: MWE09, ACL-IJCNLP (2009).
31. Tapanainen, P., Järvinen, T.: A non-projective dependency parser. In: the 5th Conference on Applied Natural Language Processing, pp. 64–71. (1997)
32. Verma, M. K. (ed.): Complex Predicates in South Asian Languages. Manohar Publications. New Delhi (1993)
33. Prince, A., Smolensky, P.: Optimality Theory: constraint interaction in generative grammar. In: Technical Report, Rutgers Center for Cognitive Science. (1993)
34. Aissen, J.: Markedness and subject choice in Optimality Theory. Natural Language and Linguistic Theory 17:673–711. (1999)
35. Bharati, A., Husain, S, Sharma, D.M., Sangal, R.: A Two-Stage Constraint Based Dependency Parser for Free Word Order Languages. In: the COLIPS IALP. (2008)
36. Bharati, A., Sangal, R., Reddy, T.P.: A Constraint Based Parser Using Integer Programming, In: ICON. (2002)
37. Bharati, A. Sangal, R.: Parsing Free Word Order Languages in the Paninian Framework. In: ACL: 93 (1993)
38. Bharati, A., Sharma, D. M., Bai, L, Sangal, R.: AnnCorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. LTRC-TR31. (2006)
39. Butt. M.: The Structure of Complex Predicates in Urdu. CSLI Publications (1995)
40. Ratnaparkhi, A.: Maximum entropy models for natural language ambiguity resolution. Ph.D. Dissertation, University of Pennsylvania. IRCS Tech Report IRCS-98-15. (1998)

SRUTHILAYA REDDY KESIDI
LANGUAGE TECHNOLOGIES RESEARCH CENTRE,
IIIT-HYDERABAD,
INDIA
E-MAIL: <SRUTHILAYA@STUDENTS.IIIT.AC.IN>

PRUDHVI KOSARAJU
LANGUAGE TECHNOLOGIES RESEARCH CENTRE,
IIIT-HYDERABAD,
INDIA
E-MAIL: <PRUDHVI@STUDENTS.IIIT.AC.IN>

MEHER VIJAY
LANGUAGE TECHNOLOGIES RESEARCH CENTRE,
IIIT-HYDERABAD,
INDIA
E-MAIL: <MEHERVIJAY.YELETI@RESEARCH.IIIT.AC.IN>

SAMAR HUSAIN
LANGUAGE TECHNOLOGIES RESEARCH CENTRE,
IIIT-HYDERABAD,
INDIA
E-MAIL: <SAMAR@RESEARCH.IIIT.AC.IN>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 73–87
RECEIVED 02/11/10 ACCEPTED 21/12/10 FINAL 16/02/11

Zero Pronominal Anaphora Resolution for the Romanian Language

CLAUDIU MIHĂILĂ,[1,⋆] IUSTINA ILISEI,[2] AND DIANA INKPEN[3]

[1] *"Al.I. Cuza" University of Iaşi, Romania*
[2] *University of Wolverhampton, UK*
[3] *University of Ottawa, Canada*

ABSTRACT

This paper presents a new study on the distribution, identification, and resolution of zero pronouns in Romanian. A Romanian corpus, including legal, encyclopaedic, literary, and news texts has been created and manually annotated for zero pronouns. Using a morphological parser for Romanian and machine learning methods, experiments were performed on the created corpus for the identification and resolution of zero pronouns.

KEYWORDS: *zero pronoun, ellipsis, anaphora resolution, Romanian, machine learning*

1 INTRODUCTION

In natural language processing (NLP), coreference resolution is the task of determining whether two or more noun phrases have the same referent in the real world [1]. This task is extremely important in discourse analysis, since many natural language applications benefit from a successful coreference resolution. NLP sub-fields such as information extraction, question answering, automatic summarisation, machine translation, or generation

⋆ The author is now with the National Centre for Text Mining, School of Computer Science, University of Manchester, 131 Princess Street, Manchester M1 7DN, UK.

of multiple-choice test items [2] depend on the correct identification of coreferents.

Zero pronoun identification is one of the first steps towards coreference resolution and a fundamental task for the development of preprocessing tools in NLP. Furthermore, the resolution of zero pronouns improves significantly the performance of more complex systems.

This paper is structured as follows: section 2 contains a description of subject ellipsis occurring in Romanian. Section 3 highlights some of the recent works in zero pronoun resolution for several languages, including Romanian. In section 4, the corpora on which this work was performed are described, and in section 5 the method is presented and the results of the evaluation are analysed.

2 ZERO SUBJECTS AND ZERO PRONOUNS

The definition of ellipsis in the case of Romanian is not very clear and a consensus has not yet emerged. Many different opinions and classifications of ellipsis types exist, as is reported in [3]. Despite the existing controversy, in this work we adopt the theory that follows.

Two types of elliptic subjects are found in Romanian: zero subjects and implicit subjects. Although both these two types are missing from the text, the difference between them is that whilst the implicit subject can be lexically retrieved, such as in example 1, the zero subject cannot, as shown in example 2.

1. $_{zp}$*[Noi]*[1] mergem la şcoală.
   [We] are going to school.
2. ∅ Ninge.
   [It] is snowing.

In Romanian, clauses with zero subject are considered syntactically impersonal, whereas implicit or omitted subjects, which are not phonetically realised, can be retrieved lexically [4].

A zero pronoun (ZP) is the gap (or zero anaphor) in the sentence that refers to an entity which provides the necessary information for the gap's correct understanding. Although many different forms of zero anaphora (or ellipsis) have been identified (e.g., noun anaphora, verb anaphora), this study focusses only on zero pronominal anaphora, which occurs when

[1] From this point forward, we denote by $_{zp}$*[]* a zero pronoun (e.g., implicit subject), whereas a zero subject will be marked using the ∅ sign.

an anaphoric pronoun is omitted but nevertheless understood [1]. An anaphoric zero pronoun (AZP) results when the zero pronoun corefers to one or more overt nouns or noun phrases in the text.

The difficulty that arises in the task of identifying zero pronouns is to distinguish between personal and impersonal use of verbs. Whilst impersonally used verbs take zero subjects (and thus have no associated ZP), personally used verbs need a subject, which in turn can be explicit or implicit. The main classes of impersonal verbs are exemplified in what follows. The examples' translation into English may sound stilted, but this is in order to provide a better understanding of the phenomenon for non-Romanian speakers.

1. Meteorological phenomena:
   ⊘ *S-a înnorat* azi. ⊘ *Este* iarnă.
   [It] clouded over today. [It] is winter.
2. Changes in the moments of the day:
   ⊘ *Se luminează* de ziuă la ora opt.
   [It] is dawning at eight o'clock.
3. Impersonal expressions with dative:
   ⊘ Îmi pare rău pentru tine. Azi ⊘ nu-mi arde de glumă.
   [It] feels sorry to me for you. Today [it] doesn't feel like joking to me.
4. Impersonal constructions with verbs *dicendi*:
   ⊘ Se vorbeşte despre ea.
   [People] are talking about her.
5. Romanian impersonal constructions with personal verbs when preceded by the reflexive pronoun "se":
   ⊘ *Se cântă* aici.
   [People] are singing here.

For the resolution step, a similar challenge exists. In this case, the main issue is to define a list of antecedent candidates, and to choose the correct one.

3 RELATED RESEARCH

Most of the studies developed for the task of coreference resolution were performed for the English language. Consequently, publicly available corpora that were created for this task are also available mostly for English, e.g., at the Message Understanding Conferences (MUC6 and MUC7[2]) [5].

[2] http://www-nlpir.nist.gov/related_projects/muc/

In the context of machine translation, a hand-engineered rule-based approach to identify and resolve Spanish zero pronouns that are in the subject grammatical position is proposed by [6]. In their study, a slot unification grammar parser is used to produce either full parses or partial parses according to a runtime parameter. The parser produced ”slot structures” that had empty slots for unfilled arguments. These were used to detect zero pronouns for verbs that were not imperatives or impersonal (e.g., ”Llueve.”/”[It] rains.”). For testing the zero pronouns, the Lexesp corpus was used, which contains Spanish texts from different genres. It has 99 sentences, containing 2213 words, with an average of 21 words per sentence. The employed heuristics detected 181 verbs, of which 75% had a missing subject, and the system resolved 97% of those subjects correctly.

Furthermore, another Spanish corpus annotated with more than 1200 ZPs was created to complement the previous study by considering the detection of impersonal clauses using hand-built rules; the reported F-measure is 57% [7, 8].

Ching-Long Yeh tried detecting and resolving zero pronouns in Chinese [9] by POS-tagging followed by phrase-level chunking. Data structures called *triples* were created from the chunked sentence, to be used both in detecting zero pronouns and in resolving them. Yeh tested on a corpus of 150 news articles containing 4631 utterances and 41000 words. A precision of 80.5% in detecting zero pronouns was reported. For the resolution stage, a recall of 70% and a precision of 60.3% of the total zero anaphors were achieved.

Converse [10] developed a rule-based approach on data from the Penn Chinese Treebank. The heuristic used is based upon Hobbs' algorithm, which traverses the surface parse tree in a particular order looking for a noun phrase of the correct gender and number [11]. Converse defined rules to substitute the lack of gender and number verb markers in the corpus, and imposed selectional/semantic restrictions, both in order to reduce the number of candidates and obtain a better accuracy. The computed recency baseline is 35%, and the top score is 43%.

Another machine learning approach which identifies and resolves zero pronouns for Chinese is described in [12], and the results are comparable to the ones obtained in [10]. Making use of parse trees and simple rules to determine the ZP and NP candidates, two classifiers are built for the identification of actual ZP from the candidate list and for resolving each of the previously identified ZPs to one of the candidate NPs. The feature vectors were then computed for the ZP candidates and for their antecedents, and

a value of 28.6% was obtained for the task of identifying zero pronouns, and 26% for resolution. However, the training data is highly disproportionate, with only one positive example for 29 negative examples. The best results were reported at a ratio of 1:8 positive:negative.

Other languages that have been more intensively and recently studied are Portuguese [13], Japanese [14] and Korean [15, 16].

In contrast, fewer studies have been performed for the coreference resolution in Romanian. A data-driven SWIZZLE-based system for multilingual coreference resolution is presented in [17]. The authors create a bilingual collection by having the MUC-6 and MUC-7 coreference training texts translated into Romanian by native speakers, and using, wherever possible, the same coreference identifiers as the English data and incorporating additional tags as needed. By using an aligned English-Romanian corpus, they exploit natural language differences to reduce uncertainty regarding the antecedents and manage to correctly resolve coreferences. Furthermore, bilingual lexical resources are used, such as an English-Romanian dictionary and WordNets, to find translations of the antecedents for each of the language.

Another study on a rule-based Romanian anaphora resolution system relying on RARE [18] was reported in [19]. First, the input is analysed using a morphological parser and a nominal group identifier. Afterwards, by employing hand-written weighted rules, such as regarding agreement in person, gender, or number, the system manages to identify coreferential chains with a success rate of 70% and an MUC precision and recall of 25% and 60%, respectively.

However, it should be noted that none of these studies consider zero pronominal anaphora in their development.

4 CORPORA

This section describes the corpora on which this study is based. In the first subsection, details about the annotation are provided, whilst in the second subsection some statistics regarding the distribution of zero pronouns in the corpora are included.

The documents included in the corpus are classified in four genres, i.e., law (LT), newswire (NT), encyclopaedia (ET), and literature (ST). The newswire texts contain international news published in the beginning of 2009, while the law part of the corpus represents the Romanian constitution. The literary part is composed of children's short stories by Emil

Gârleanu and Ion Creangă, whilst the encyclopaedic corpus comprises articles from the Romanian Wikipedia on various topics.

The important contribution of this study is two-fold: the selection of genres which are likely to be relevant to several NLP applications (e.g., multiple choice test generation, question answering), and manual annotation of all four genres with the anaphoric zero pronouns information.

In what follows, the annotation setup is provided, and some statistics regarding the distribution of zero pronouns are presented in the second subsection.

4.1 *Annotation*

The documents comprised in the corpora were parsed automatically using the web service published by the Research Institute for Artificial Intelligence[3], part of the Romanian Academy. This parser provides the lemma and the morphological characteristics regarding the tokens.

The texts were afterwards manually annotated for zero pronouns by two authors, in order to create a golden standard. The inter-annotator agreement regarding the existence of zero pronouns is 90%.

A zero pronoun was manually identified by the addition of the following empty XML tag containing the necessary information as attributes into the parsed text:

```
<ZERO_PRONOUN id="w152.5" ant="w136"
depend_head="w153" agreement="high"
sentence_type="main" />
```

Each ZERO_PRONOUN tag includes various pieces of information regarding its antecedent (the `ant` attribute), the verb it depends on (the `depend_head` attribute) and the type of sentence it appears in (i.e., the `sentence_type` attribute). The attribute coresponding to the antecedent may have one of three types of values: (i) *elliptic*, if there is no antecedent, (ii) *non_nominal*, if the antecedent is a clause, or (iii) a unique identifier which points back to the antecedent, in the case of an AZP. The dependency head attribute points to the verb on which the zero pronoun depends. If the verb is complex, it points to the auxiliary verb. In order to cover the possible clauses where the zero pronoun appears, one more attribute (sentence type) provides information about the kind of sentence (main, coordinated, subordinated, etc.).

[3] http://www.racai.ro/webservices/

4.2 *Statistics*

The currently gathered corpus comprises over 55000 tokens and almost 1000 zero pronouns, as shown in Table 1. Nevertheless, it can be noticed from the table that the legal and literary texts have a very low and a very high, respectively, density of ZP per sentence.

Table 1. Description of the corpora.

Overview	ET	LT	ST	NT	Overall
No. of tokens	17191	13739	5141	19374	55445
No. of sentences	728	790	371	852	2741
No. of ZP	235	113	391	258	997
Avg. tokens/sentence	23.61	17.39	13.85	22.73	20.22
Avg. ZP/sentence	0.32	0.14	1.05	0.30	0.36

The distribution of the zero pronouns in the studied corpora is provided in Table 2. The distances from zero pronouns to their antecedents in the case of newswire and literature texts reveal unique values. This is due to the different writing styles, in which either to avoid possible misinterpretations, or to increase the fluency of narrative sequences, the authors adjust the use of zero pronouns. However, the distance to the dependent verb is constant throughout the corpora, which is on average 1.68 tokens away.

Table 2. Distances between the ZP and its antecedent and dependent verb.

Corpus	Antecedent (sentences)	Antecedent (tokens)	Dependent verb (tokens)
ET	0.68	23.87	1.77
LT	1.07	38.55	1.56
ST	2.92	46.63	1.62
NT	0.02	7.44	1.74
Overall	1.43	30.21	1.68

Considering that no previous study has been undertaken for the Romanian language, we note that the results for the encyclopaedic and legal texts can be compared to the ones obtained for another Romance language, Spanish, in [7].

5 EVALUATION

5.1 *Identification of Zero Pronouns*

The first goal is to classify the verbs into two distinct classes, either with or without a zero pronoun. The chosen method in this study is supervised machine learning, using Weka[4] [20, 21]. Therefore, a feature vector was constructed for the verbs. The vector is composed of the following eleven elements:

- type – the type of the verb (i.e. main, auxiliary, copulative, or modal);
- mood – the mood of the verb (indicative, subjunctive, etc.);
- tense – the tense of the verb (present, imperfect, past, pluperfect);
- person – the person of the conjugation (first, second, or third);
- number – the number of the conjugation (singular or plural);
- gender – the gender of the conjugation (masculine, feminine, neuter);
- clitic – whether the verb appears in a clitic form;
- impersonality – whether the verb is strictly impersonal (such as meteorological verbs);
- 'se' – whether the verb is preceded by the reflexive pronoun "se";
- number_of_verbs_in_sentence – the number of verbs in the sentence where the candidate verb is located;
- hasZP – whether the verb has a ZP.

The first seven elements of the feature vector are extracted from the morphological parser's output, whilst the next three elements are computed automatically based on the annotated texts. The last item is the class whose values are true if the verb allows zero pronouns and false otherwise, and it is used only for training purposes. When in test mode the class is not used, except when computing the evaluation measures.

The data set on which the experiments were performed includes 1994 instances of the feature vector. Half of these instances correspond to the 997 verbs which have an associated ZP, whilst the other half contains randomly selected verbs without a ZP. As the baseline classifier employed, ZeroR, takes the majority class, the baseline to which we need to compare our accuracy is 50%.

We experimented with multiple classifiers pertaining to different categories. The results that follow are obtained by 10-fold cross validation on the data. Precision, Recall and F-measure for each of the classes of

[4] http://www.cs.waikato.ac.nz/ml/weka/

verbs and the accuracy for three classifiers (SMO, Jrip, and J48) and one meta-classifier (Vote) are included in Table 3. SMO is the implementation of SVM, J48 is an implementation of decision trees, and Jrip is an implementation of decision rules. The Vote meta-classifier is configured to consider the three previous classifiers using a Majority Voting combination rule.

Table 3. Scores from four classifiers for the classes of verbs.

Classifier	Accuracy	has ZP			not ZP		
		P	R	F_1	P	R	F_1
SMO	0.739	0.814	0.620	0.704	0.693	0.859	0.767
Jrip	0.734	0.722	0.764	0.742	0.749	0.705	0.727
J48	0.746	0.783	0.683	0.730	0.719	0.810	0.762
Vote	0.745	0.785	0.675	0.726	0.715	0.815	0.762

The results may vary slightly, since only a subset of verbs with no ZP was selected. Nevertheless, repetitions of the experiment with different test datasets produced similar values. As observed, the Vote meta-classifier does not improve the results, which leads us to the conclusion that the three classifiers make relatively the same decisions.

In order to observe the rules according to which the decisions are made, the Jrip classifier was employed. The obtained output is included in Figure 1. The most used attribute is clearly the mode of the verb, whilst the gender and the clitic form do not appear at all.

Aiming at determining which features most influence classification, regardless of the classifying algorithm, two attribute evaluators have provided the results shown in Table 4.

As expected, the most problematic case is that of the present indicative verbs in the third person and preceded by the reflexive pronoun ”se”. A reason for this effect is that ”se” is part of impersonal constructions which may or may not have zero pronouns. As a result, the system classifies the verbs incorrectly.

5.2 *Resolution of Zero Pronouns*

The second goal of our research is to find the correct antecedent to resolve the anaphor. The methodology employed in resolving zero pronouns is supervised machine learning, using the aforementioned gold corpus as training and test data.

```
(MOOD = 1) => HASZP = true (308.0/33.0)
(MOOD = 0) and (TENSE = 2)
    => HASZP = true (200.0/58.0)
(MOOD = 0) and (VERBNUMBERINSENTENCE >= 6)
    => HASZP = true (140.0/44.0)
(MOOD = 0) and (TENSE = 3) => HASZP = true (28.0/2.0)
(MOOD = 0) and (PERSON = 0)
    => HASZP = true (36.0/6.0)
(PERSON = 2) and (VERBNUMBERINSENTENCE >= 5) and
                 (VERBNUMBERINSENTENCE >= 6)
    => HASZP = true (139.0/58.0)
(MOOD = 0) and (NUMBER = 0) and (TENSE = 1)
    => HASZP = true (52.0/14.0)
(MOOD = 0) and (NUMBER = 0) and (PERSON = 1)
    => HASZP = true (22.0/3.0)
    => HASZP = false (1069.0/290.0)
```

Figure 1: Rules output from the Jrip classifier.

Table 4. Attribute selection output from two attribute evaluators.

Attribute	ChiSquare	InfoGain
Mood	437.09	0.1728
Person	29.09	0.0108
Verb number in sentence	15.97	0.0058
Tense	15.42	0.0057
Type	14.65	0.0053
Impersonality	10.35	0.0044
Number	7.44	0.0026
'Se'	5.28	0.0019
Gender	1.95	7E-4
Clitic	0	0

The feature vector that was constructed for the verbs and antecedent candidates is composed of 21 elements, the first nine of which are the same as in the identification stage. The other are briefly described in what follows:

- number_of_verbs_in_sentence – the number of verbs in the sentence where the zero pronoun is located;
- candidate_pos – the part of speech of the candidate (i.e. noun or pronoun);

- candidate_type – the type of the candidate (i.e. main, auxiliary, copulative, or modal);
- candidate_case – the case of the candidate (direct, oblique, or vocative);
- candidate_person – the person of the candidate (first, second, or third);
- candidate_number – the number of the candidate (singular or plural);
- candidate_gender – the gender of the candidate (masculine, feminine, or neuter);
- candidate_definite – whether the candidate appears in a definite form;
- candidate_clitic – whether the candidate appears in a clitic form;
- distance_sentences – the distance in sentences between the verb and the candidate;
- distance_tokens – the distance in tokens between the verb and the candidate;
- isAnt – whether the candidate is the ZP's antecedent (verb's subject).

Two baselines have been taken into account for this stage. Firstly, the ZeroR classifier takes the majority class as the class for the entire population. Due to the selection of the data, its accuracy is 50%.

The second baseline employed considers as antecedent the first previous noun, pronoun, or numeral which is in gender and number agreement with the verb. Its accuracy is really low, only 12.52%. Most of the cases that are correctly identified by this baseline are those in which the verb is in the subjunctive mood, and the antecedent precedes it and is declined in the oblique case. Such an example is included in the sentence below, where *it* refers to *Macedonia*.

[...] a cerut Macedoniei $_{zp}$*[ea]* să stabilească relaţii diplomatice la Kosovo.
[...] asked Macedonia $_{zp}$*[it] to establish diplomatic relations in Kosovo.*

The classifiers that were experimented with are the same as those in the prior identification stage. The SMO, JRip, and J48 classifiers and Vote meta-classifier were run with a 10-fold cross validation, and the results that were obtained are included in Table 5.

Due to the fact that only a subset of false candidates was considered in the training and test data, the results vary between various re-runs of the experiment. However, repeating the experiment several times with different data proved that the variations are small and are not statistically significant. The SVM classifier is outperformed by the other two, decision trees and decision rules, and also by the Vote meta-classifier.

Figure 2 shows decision rules for the JRip classifier. The features that occur on higher levels, such as the distances, candidate case, POS, or definiteness, appear to help classify most of the given antecedents.

Table 5. Classifier results for the classes of candidates.

Classifier	Accuracy	is Antecedent			not Antecedent		
		P	R	F_1	P	R	F_1
SMO	0.727	0.717	0.751	0.733	0.738	0.703	0.720
JRip	0.839	0.882	0.783	0.829	0.805	0.895	0.848
J48	0.864	0.852	0.882	0.867	0.877	0.847	0.862
Vote	0.865	0.867	0.862	0.865	0.863	0.868	0.865

```
(DISTANCESENTENCES >= 1) and (DISTANCESENTENCES <= 5)
   and (CANDIDATEDEFINITE = 1)
   => ISANTECEDENT = false (372.0/28.0)
(DISTANCESENTENCES >= 1) and (DISTANCESENTENCES <= 5)
   and (VERBNUMBERINSENTENCE >= 4)
   => ISANTECEDENT = false (202.0/28.0)
(DISTANCESENTENCES >= 1) and (DISTANCESENTENCES <= 5)
   and (CANDIDATECASE = 1)
   => ISANTECEDENT = false (68.0/8.0)
(DISTANCESENTENCES >= 1) and (DISTANCESENTENCES <= 5)
   and (CANDIDATENUMBER = 1)
   => ISANTECEDENT = false (45.0/7.0)
(DISTANCESENTENCES >= 1) and (DISTANCESENTENCES <= 5)
   and (DISTANCESENTENCES >= 2) and (CANDIDATEPOS = 0)
   => ISANTECEDENT = false (166.0/57.0)
(CANDIDATEPOS = 1) and (CANDIDATEDEFINITE = 1)
   => ISANTECEDENT = false (37.0/1.0)
(CANDIDATETYPE = 5)
   => ISANTECEDENT = false (13.0/3.0)
(DISTANCESENTENCES >= 1) and (DISTANCESENTENCES <= 3)
   and (CANDIDATETYPE = 0)
   => ISANTECEDENT = false (9.0/1.0)
(CANDIDATECASE = 1) and (DISTANCETOKENS >= 16)
   => ISANTECEDENT = false (4.0/0.0)
   => ISANTECEDENT = true (824.0/87.0)
```

Figure 2: Rules output from the Jrip classifier.

The attributes that are the most salient in this classification, according to Table 6, are the distances between the candidate and the verb, measured in both sentences and tokens. Other very important attributes are the definiteness, case, and type of the candidate, as can also be observed from the aforementioned decision rules.

It is important to note that the learning model relies more on candidate features than on verb features. While some of the candidate features have very high values, most of the verb features are given a null value by the two attribute evaluators, ChiSquare and InfoGain. The features with null values have been omitted from the table.

Table 6. Resolution attribute selection output from two attribute evaluators.

Attribute	ChiSquare	InfoGain
Distance in sentences	770.282	0.3608
Distance in tokens	491.870	0.2212
Candidate definite	168.496	0.0714
Candidate case	154.011	0.0688
Candidate type	93.328	0.0480
Verb number in sentence	20.063	0.0083
Candidate person	7.825	0.0041
Candidate gender	5.542	0.0022
Verb mood	5.062	0.0021
Verb type	1.447	0.0006
Candidate PoS	1.100	0.0004
Verb tense	0.819	0.0003
Candidate number	0.224	0.0001

6 CONCLUSIONS AND FUTURE WORK

This paper presents a study on the distribution, identification, and resolution of zero pronouns in Romanian. By creating and manually annotating a multiple-genre corpus, zero pronouns are identified and resolved using supervised machine learning algorithms. The accuracies of 74% for identification and 86% for resolution are comparable to those obtained for other languages for which such studies have been performed.

Concerning the usability of this study, applications include question answering and automatic summarisation. As a large number of ZPs are present in text, extracting the correct subject of important actions is vital. Furthermore, machine translation might benefit for pairs of languages with different rules regarding zero pronouns. Moreover, since the distribution depends largely on the genre, it might depend on the author as well, and thus automatic zero pronoun identification might be used in plagiarism and authorship detection.

REFERENCES

1. Mitkov, R.: Anaphora Resolution. Longman, London (2002)
2. Mitkov, R., Ha, L.A., Karamanis, N.: A computer-aided environment for generating multiple-choice test items. Journal of Natural Language Engineering **12** (2006) 177–194
3. Mladin, C.I.: Procese şi structuri sintactice ”marginalizate” în sintaxa românească actuală. Consideraţii terminologice din perspectivă diacronică asupra contragerii - construcţiilor - elipsei. The Annals of Ovidius University Constanţa - Philology **16** (2005) 219–234
4. Institutul de Lingvistică ”Iorgu Iordan - Al. Rosetti” Bucureşti: Gramatica limbii române. Editura Academiei Române, Bucureşti (2005)
5. MUC 7: Proceedings of the 7^{th} Message Understanding Conference. Morgan Kaufmann Publishers (1998)
6. Ferrández, A., Peral, J.: A computational approach to zero-pronouns in Spanish. In: ACL ’00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2000) 166–172
7. Rello, L., Ilisei, I.: A comparative study of Spanish zero pronoun distribution. In: Proceedings of the International Symposium on Data and Sense Mining, Machine Translation and Controlled Languages (ISMTCL). (2009)
8. Rello, L., Ilisei, I.: A rule based approach to the identification of Spanish zero pronouns. In Temnikova, I., Nikolova, I., Konstantinova, N., eds.: Proceedings of the Student Workshop at RANLP 2009. (2009) 60–65
9. Yeh, C.L., Chen, Y.C.: Zero anaphora resolution in Chinese with shallow parsing. Journal of Chinese Language and Computing **17** (2007) 41–56
10. Converse, S.P.: Pronominal anaphora resolution in Chinese. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA (2006)
11. Hobbs, J.R.: Resolving pronoun references. Lingua **44** (1978) 311–338
12. Zhao, S., Ng, H.T.: Identification and Resolution of Chinese Zero Pronouns: A Machine Learning Approach. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics (2007) 541–550
13. Pereira, S.: ZAC.PB: An annotated corpus for zero anaphora resolution in Portuguese. In Temnikova, I., Nikolova, I., Konstantinova, N., eds.: Proceedings of the Student Workshop at RANLP 2009. (2009) 53–59
14. Iida, R., Inui, K., Matsumoto, Y.: Exploiting syntactic patterns as clues in zero-anaphora resolution. In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2006) 625–632
15. Kim, Y.J.: Subject/object drop in the acquisition of Korean: A cross-linguistic comparison. Journal of East Asian Linguistics **9** (2000) 325–351

16. Han, N.R.: Korean zero pronouns: analysis and resolution. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA (2006)
17. Harabagiu, S.M., Maiorano, S.J.: Multilingual coreference resolution. In: Proc. of 6^{th} conference on Applied natural language processing, Morristown, NJ, USA, Association for Computational Linguistics (2000) 142–149
18. Cristea, D., Postolache, O.D., Dima, G.E., Barbu, C.: AR-Engine – a framework for unrestricted co-reference resolution. In: Proceedings of the LREC 2002, Third International Conference on Language Resources and Evaluation. (2002) 2000–2007
19. Pavel, G., Postolache, O., Pistol, I., Cristea, D.: Rezoluţia anaforei pentru limba română. In Forăscu, C., Tufiş, D., Cristea, D., eds.: Lucrările atelierului Resurse lingvistice şi instrumente pentru prelucrarea limbii române, Iaşi (2006)
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations **11** (2009)
21. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann (2005)

CLAUDIU MIHĂILĂ
FACULTY OF COMPUTER SCIENCE,
"AL.I. CUZA" UNIVERSITY OF IAŞI,
16 GENERAL BERTHELOT STREET, IAŞI 700483,
ROMANIA
E-MAIL: <CLAUDIU.MIHAILA@CS.MAN.AC.UK>

IUSTINA ILISEI
RESEARCH INSTITUTE IN
INFORMATION AND LANGUAGE PROCESSING,
UNIVERSITY OF WOLVERHAMPTON,
WULFRUNA STREET, WOLVERHAMPTON WV1 1LY,
UK
E-MAIL: <IUSTINA.ILISEI@GMAIL.COM>

DIANA INKPEN
SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING,
UNIVERSITY OF OTTAWA,
800 KING EDWARD STREET, OTTAWA, ON, K1N 6N5,
CANADA
E-MAIL: <DIANA@SITE.UOTTAWA.CA>

Semantics and Discourse

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 91–103
RECEIVED 01/11/10 ACCEPTED 02/12/10 FINAL 08/02/11

Recognizing Deverbal Events in Context

IRENE RUSSO, TOMMASO CASELLI, AND FRANCESCO RUBINO

Istituto di Linguistica Computazionale, CNR, Italy

ABSTRACT

Event detection is a key task in order to access information through content. This paper focuses on events realized by deverbal nouns in Italian. Deverbal nouns obtained through transpositional suffixes (such as -zione; -mento, -tura and -aggio) are commonly known as nouns of action, i.e. nouns that denote the process/action described by the corresponding verbs. However, this class of nouns is also known for a specific polysemous alternation: they may denote the result of the process/action of the corresponding verb. This paper describes a statistically based analysis that helps to develop a classifier for automatic identification of deverbal nouns denoting events in context by exploiting rules obtained from syntagmatic and collocational cues identified by linguists.

1 INTRODUCTION

In Italian, deverbal nouns obtained through transpositional suffixes (such as *-zione*; *-mento*, *-tura*; and *-aggio*) are commonly known as nouns of action (*nomina actionis*) or nominalizations, i.e. nouns which denote the process/action described by the corresponding verbs. This class of nouns is also known for a specific lexical ambiguity phenomenon: they may denote the result of the process/action of the corresponding verbs. Commonly, these two different denotations of deverbal nouns are named *event* (example 1) and *result* (example 2) reading:

(1) La **costruzione**$_{EVENT}$ del ponte é durata tre anni.
The building of the bridge lasted three years.

(2) Questa *costruzione*$_{RESULT}$ é imponente.
This building is huge.

This paper focuses on a statistically based analysis for the disambiguation of Italian deverbal nouns in context, using syntagmatic and collocational cues that are specific for the identification of the eventive reading. The classifier has been built using J48, the rule version of the decision tree classifier C4.5, and distributed through the Weka platform [1]. Two different training sets have been used: the It-TimeBank corpus[1], a corpus of Italian newpaper articles annotated with the Italian version of the TimeML specifications [2] and the La Repubblica Corpus [3].

From each set of data we have extracted co-occurence frequencies with a list of relevant syntagmatic cues (namely verbs and adjectives) identified through a detailed review of linguistically oriented works such as [4–6].

Next to this set of linguistically informed cues, we have also experimented the use of parts-of-speech (POS) sequences, which from previous works in word sense disambiguation tasks have proved useful ([7] among others).

In addition to the development of a classifier for disambiguating the eventive reading of deverbal nouns in Italian, we also want to verify the usefulness of the linguistically informed features, i.e. how powerful they are in discriminating the correct reading, by exploiting different types of training data, namely manually annotated tokens (single sentence level) *vs*. distributional frequencies of pre-classified types (global corpus level). We test whether a combination of relevant lexical cues useful for broad semantic classification out of context and syntactic patterns essential for the discrimination in context can help for the disambiguation of deverbal nouns.

The remaining of the paper is structured as follows: Section 2 describes the set of linguistically motivated features that emerges from the review of previous related works. Section 3 is devoted to the description of the classifier by means of the the experiments we conducted and their evaluation. In Section 4 the methodology we have adopted is compared with previsous works in NLP on this subject. A tentative comparison of the results is outlined though the data sets used for the evaluation are different. Finally, Section 5 reports on the conclusions and future developments.

[1] The corpus is still under development and not officially distributed.

2 SYNTAGMATIC AND LEXICAL CUES FOR DEVERBAL NOUNS DISAMBIGUATION

To automatically detect nouns that denote an event, morphological suffixation provides an important cue. However, deverbal nouns exhibit a peculiar and complex kind of logical polysemy [8]. The deverbal nouns can denote the act but also the result of an action, both as an abstract result (as in "*L'**espressione** di Maria fu inopportuna*" [Maria's statement was out of context]) or as a concrete result (as in "*L'**espresione** scritta alla lavagna era scorretta*" [The formula on the blackboard was wrong]). In these cases, the new meaning is the object of the verb though in other cases the non-eventive meaning can denote a result state ("*la **coagulazione** del sangue*" [blood clotting]), an instrument ("*L'**illuminazione** della sala fu rimessa in funzione*" [The illumination of the room was brought back into operation]), a material ("*la **segatura***" [the sawdust]), a person or object responsible for the action ("*la **difesa** accusò i giudici*" [the defense accused the judges]), the place where the predicate is realized ("*la sua **sistemazione** era un lussuoso appartamento*" [his accommodation was a luxury apartment]), the modality ("*la **classificazione** dei libri è pessima*" [the book classification is wrong]).

Theorical literature on this subject such as [5, 9, 6] points out the selection of specific cues for the identification of the two possible readings. For clarity's sake we report in Table 1 this set of cues.

Table 1. Cues for the identification of the eventive *vs*. non-eventive reading of deverbal nouns.

Features/cues	**Event reading**	**Non-eventive reading**
Obligatory realization of verb argument structure by means of a PPs	+	-
Pluralization	+	-
Telicity of the verb	+	-
Verb grammatical class	+	+
Type of determiner	+	-
Aspectual modifiers	+	-
Agent-oriented modifiers	+	-
Co-occurrence with eventive predicate	+	-
Complement clause at the infinitive	+	-
by-phrases, relational adjectives and possessive determiners as realizations of the subject of the deverbal noun	+	-

In [10], the relevance of these cues was verified through a corpus-based quantitative analysis on a set of 842 Spanish deverbal nouns (over a total of 3,075 occurrences and 1,121 senses). We claim that their results can be applied to Italian given the high similarity of the two languages.

Among the scholars there is not a complete agreement on these features. Moreover, a small set of cues is suggested but no effort is made to establish the nature of their discriminative role (i.e are they dichotomous?) or to rank the cues on the basis of their discriminative strength. As a consequence, linguistic theories lack of classification rules that instead are strictly necessary for computational systems. The identification of the most relevant cues and corresponding values must be carefully conducted since we aim at automatically detect them in text.

In the remaining of this section, we will go through some of the features listed in Table 1. Descriptive statistics on the features are reported to briefly asses their import. The figures have been obtained through a test set of 581 deverbal nouns extracted from the It-TimeBank (see Section 3 for details on the resource) Corpus. The test set contains 440 occurrences of eventive nouns and 141 non-eventive nouns.

Obligatory realization of the argument structure with a PP One of the most controversial point is related to the role of argument structure. [5] claims that only complex event nouns[2] have argument structure and its realization is compulsory. On the other hand, other scholars ([9], [6] among others) consider the presence of argument structure as an ancillary element for the disambiguation of deverbal nouns. These authors go even further in claiming that all event nouns, both complex and simple, can have arguments and its overt (i.e. superficial) realization is not necessary in order to instantiate the event reading.

For instance, the noun "*fucilazione*" [shooting] has event readings both in example 3 and in 4, though in 4 there is no overt (superficial) argument realization:

(3) La **fucilazione**$_{EVENT}$ della prigioniera$_{Arg1}$ da parte dei soldati$_{Arg0}$.
The shooting of the prisoner by the soldiers.

(4) La **fucilazione**$_{EVENT}$ ha avuto luogo nella piazza.
The shooting took place in the square.

[2] In her account, Grimshaw distinguishes among three types of nominalizations, namely (i.) complex event nominals, which requires the obligatory realization of the verb argument structure, (ii.) simple event nominals, which have event reading but do not realize argument structure and (iii.) result nouns.

The results from [10]'s analysis have provided a partial support to this latter hypothesis. They have observed that almost every eventive reading of deverbal nouns (98%) presents a realization of the argument structure. However, they have also observed that there are cases in which the argument structure is not realized and argument structure can be realized by constituents other than PPs, such as possessive determiner. As for Italian [8] argues that predicate arguments can be omitted but they are frequently expressed through the preposition "*di*" and that possessive adjectives can express arguments as well.

On the basis of these results, the presence of the argument structure could be a discriminating cue but the automatic detection of internal arguments of deverbal nouns is not an easy task due to the fact that their identification is subordinated to the identification of the status of the deverbal noun (eventive *vs*. non-eventive). In Table 2 we report the percentages of nouns co-occurring with the realization of the argument structure in the dataset. If the argument structure is preferentially realized through the PPs "*di/del*", it is apparent that eventive nouns are more often followed by this kind of phrases with respect to non eventive nouns. From the data, it seems that possessive modifiers tend to co-occur with non-eventive readings against linguists' intuitions.

Table 2. Co-occurrence percentages of the cues for argument structure realization.

Noun type	**Possessive modifiers**	**PPs**	**Di / Del**
eventive deverbal nouns	0.8%	47%	40%
non eventive deverbal nouns	2.5%	28%	22%

Pluralization The occurrence in plural forms of a deverbal noun is considered as a discriminating cue for detecting its non-eventive reading. As a matter of fact, [10] reports that 98% of the plural instances of deverbal nouns have a non eventive reading. On our dataset (see Table 3): deverbal eventive nouns are less frequently pluralized with respect to non-eventive nouns, even if the difference is not striking.

Type of determiner According to the theoretical literature, if the determiner of a deverbal noun is a definite article, the noun will have an eventive reading. As reported in [10], this hypothesis is not verified by a corpus analysis. Demonstratives tend to prefer resultative (i.e. non-eventive)

Table 3. Percentages of singular and plural occurrences.

Noun type	Singular	Plural
eventive deverbal nouns	87%	13%
non eventive deverbal nouns	59%	41%

readings. Even if these features are reported in literature, it is hard to define how they can be used to disambiguate the correct reading of the deverbal nouns because the differences in percentages between eventive and non-eventive readings are not significant. However, they are retained in our analysis because linguists' intuitions report on their role.

Table 4. Co-occurrence percentages of determiners.

Noun type	il/la	un/una	demonstrative
eventive deverbal nouns	39%	13%	1%
non eventive deverbal nouns	33%	10%	3.8%

Aspectual modifiers [10] did not report any figures on collocational cues in their study. However, it is possible to identify a rich list of relevant lexical items which could help in the classification of eventive nouns out of context and their identification in context. We manually selected a set of 53 high frequency adjectives and 41 verbs that can be reputed good collocational cues for the identification of eventive readings. In particular, we focus our attention on a selection of aspectual concurrent adjectives (e.g. "*annuo*" [yearly], "*contemporaneo*" [contemporary], "*immediato*" [immediate]) that modify more frequently eventive nouns. Other potentially interesting lexical cues are agent-oriented adjectives (e.g. "*abile*" [able], "*moderato*" [moderate], "*volontario*" [voluntarily])) that tend to co-occur with eventive nouns. Finally, we consider the co-occurrences of the nouns either as the object or as the subject of eventive predicates, such as "*continuare*" [to continue], "*finire*" [to finish], "*rimandare*" [to postpone] and so on and so forth.

3 TOWARDS THE CLASSIFIER: EXPERIMENTS AND RESULTS

In the development of the classifier we want to compare, on one hand, syntagmatic and collocational information from manually annotated cor-

pora with co-occurrence frequencies from large corpora extracted after a coarse grained annotation derived from a lexical resource. On the other hand, we want to test the relevance of the cues suggested by linguists with similar cues extracted without previous assumptions.

The data set we have used to train the Weka version of the C4.5 algorithm is composed by three different sets of data: two training datasets, the It-TimeBank corpus and the La Repubblica Corpus [3], and one test set, composed by a TimeML-compliant manually annotated data from the La Repubblica Corpus.

The It-TimeBank is an Italian corpus composed by 149 newspaper articles, for a total of more than 63 thousand tokens, with 18,312 of them being labelled as nouns. Six annotators have manually applied the TimeML specifications [2] by distinguishing between temporal expressions, events and signals. As far as the event annotation is concerned the corpus contains 8,138 tokens annotated as events (including verbs, nouns, adjectives and prepositional phrases), 3,695 of whom are realized by nominal tokens. As already stated, we have a grand total of 581 deverbal tokens realized by means of transpositional suffixes, which count 440 event tokens and 141 non-eventive ones. Inter-annotator agreement on event annotation is K = 0.87 and average precision and recall 0.89, which guarantee a reliable supervised data set. A subset of 31,000 tokens of this corpus was released for the SemEval 2010 TempEval-2 task [11].

The La Repubblica set is a training dataset composed by 1054 high frequency nouns and subdivided in two sub-sets: 566 deverbal nouns exclusively eventive such as "*pulitura*" [cleaning], "*proliferazione*" [proliferation] selected according to the transpositional suffixes in analysis, and 488 non eventive nouns such as "*aula*" [classroom], "*testo*" [text]. These nouns have been extracted automatically by associating to each noun in the corpus its highest hyperonym in MultiWordNet [12].

As test set we have 444 sentences randomly extracted from La Repubblica corpus containing a deverbal noun. They were manually annotated by the authors: 281 sentences contain an eventive occurrence of a deverbal noun and 163, non-eventive. The features' extraction was automatically performed on a dependency parsed version of the datasets [13].

3.1 *Experiment 1: type occurrences and token occurrences of eventive and non-eventive nouns*

We apply the J48 classifier provided by Weka, with La Repubblica data as training set. Distributional patterns have been largely used to find seman-

Table 5. Results. Here A is accuracy, P precision, R recall, and F is F-measure.

Noun type	**La Repubblica**				**It-TimeBank**			
	A	P	R	F.	A	P	R	F
event reading		0.72	0.88	0.80		0.63	1	0.77
not-event reading		0.68	0.41	0.51		0	0	0
event + not-event	71.5%	0.70	0.71	0.69	63.5%	0.40	0.63	0.49

tically related nouns at type level in large corpora [14] and have proved their utility for semantic classification tasks. For instance, [15] obtain an accuracy of 75% for the classification of eventive nouns. But the reverse is true: from previously classified semantic items discriminative distributional patterns for token occurrences can be induced.

We have considered as baseline the most frequent class as the correct one, i.e the eventive reading, which corresponds to the 63.2%. The accuracy obtained against the test set is 71.5%, which ouperforms the baseline of 8 points, with an overall F-measure of 0.69. If we split the results on the basis of the readings, or classes, of the deverbal nouns, the results show that the classifier performs better on eventive readings (F-measure = 0.80) than on non-eventive ones (F-measure = 0.51). Detailed results are reported in Table 5 under the heading "*La Repubblica*".

3.2 *Experiment 2: token occurrences as training*

The second experiment uses the It-TimeBank corpus as training set. The results are lower than those obtained when using the La Repubblica Corpus. We obtain an overall accuracy of 63.5% (F-measure = 0.49), which is very close to the baseline. It is striking to observe how with this highly supervised training set the classifier performance is worse. In particular, no non-eventive reading of the nouns in the test test is correclty classified. The details are reported again in Table 5 under the heading "*It-TimeBank*".

3.3 *Experiment 3: POS sequences as disambiguating cues*

Event noun detection for event extraction systems is partially akin to word sense disambiguation: the aim is to test algorithms for automatic detection/identification of nouns denoting events in context. Methodologies that proved their utility for WSD tasks can be tested on event nouns detection in context. For this reason, we evaluate the relevance of single

POS preceding or following our key words, performing classifications even on the basis of sequences of POS, a methodology that [7] reputed partially good for WSD of nouns. More generally, our aim is to test the role of POS sequences as not theoretically predetermined features that are similar, in terms of structural information, to more specific patterns listed by linguists. The results are discouraging (see Table 6), showing that even wider POS sequences as 5-grams are not able to help in this classification task.

Table 6. POS *n*-grams as disambiguating cue.

POS sequence	**Accuracy - La Repubblica**	**Accuracy - It-TimeBank**
P-1, P0, P+1	41%	70%
P0, P+1, P+2	63.2%	63.2%
P-2, P-1, P0	63.2%	63.2%
P-2, P-1, P0, P+1, P2	37.3%	63.2%

4 RELATED WORKS

In recent years there has been an increasing interest in the NLP community for automatic event identification as the development of different systems for the identification of event nouns shows [16], [17] [18] [19] among others).

To the best of our knowledge, our methodology (and the results obtained) can be directly compared with [20], even if they did not focus specifically on deverbal nouns. They propose a weakly-supervised method for detecting nominal events mentions that classify noun phrases on the basis of a combination of word sense disambiguation and lexical acquisition techniques. Our training and test sets are smaller but we show how, with a list of linguistically informed cues, our methodology slightly outperforms their results for eventive reading of deverbal nouns (88% *vs*. 87.7%) while is lower for non-eventive ones (41% *vs*. 60%).

Finally, comparing our results with [19] is not possible because precision and recall are reported for the component of the classifier that integrates information from a lexical resource with information extracted from a corpus. Using just corpus data, as we did in our experiments, they report an accuracy of 80%, which yields an accuracy which is lower than their baseline (82.1%). In the overall, our classifier seems to be better for

the classification of eventive readings of deverbal nouns, while it seems less promising for the classification of non eventive nouns. This may be due also to the fact that the features' set we have identified is mainly focused on eventive readings.

5 CONCLUSION AND FUTURE WORKS

The availability of methodologies able to identify the correct denotation of deverbal nouns is essential because it can help to build better event extraction system but it can also improve the performance of more complex NLP systems such as anaphora resolution, subcategorization frames, paraphrase detection and temporal processing. Our classifier can be integrated in a broader event extraction system for Italian but it can be used also for automatically annotate or add semantic information to large corpora reducing the manual effort and costs for their realization.

In this paper we show how to classify deverbal nouns in context as eventive or non eventive using syntagmatic and collocational information relative to past encounters of nouns tagged with the help of a lexical resource such as MultiWordNet.

We have showed that linguistically informed syntagmatic and lexical patterns perform better than POS sequences, at least for this task.

Future work will focus on automatic identification of nouns denoting events, going beyond the present case study on deverbal nouns. Of course, some integrations in the fatures' set to improve the identification of non-eventive readings are necessary, together with a more detailed classification of these occurrences (e.g. result/state *vs*. concrete object). The role of manually annotated data as training set such as the It-TimeBank is not clear due to its dimension with respect to the class of deverbal nouns. With a richer training set manually annotated we will gain clearer evidence on the utility of annotated corpora.

ACKNOLEDGMENTS This work has been supported by the EU founded project PANACEA (Grant agreement no.: 7FP-ITC-248064).

REFERENCES

1. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)

2. Pustejovsky, J., Castao, J., Ingria, R., Saurì, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust specification of event and temporal expressions in text. In: Fifth International Workshop on Computational Semantics (IWCS-5). (2003)
3. Baroni, M., Bernardini, S., Comastri, F., Piccioni, L., Volpi, A., Aston, G., Mazzoleni, M.: Introducing the "la Repubblica" corpus: A large, annotated, TEI(XML)-compliant corpus of newspaper italian. In: Proceedings of the Fourth International conference on Language Resources and Evaluation (LREC-04). (2004)
4. Vendler, Z.: 4. In: Linguistics in Philosophy. Cornell University Press, Ithaca, NY (1967)
5. Grimshaw, J.: Argument Structure. MIT Press, Cambridge, Massachusetts (1990)
6. Alexadiou, A.: The Functional Structure in Nominals. Nominalization and Ergativity. John Benjamins, Amsterdam/Philadelphia (2001)
7. Mohammad, S., Pedersen, T.: Combining lexical and syntactic features for supervised word sense disambiguation. In: Proceedings of the Conference on Computational Natural Language Learning. (2004) 25–32
8. Gaeta, L.: Nomi d'azione. In Grossmann, M., F., R., eds.: La formazione delle parole in italiano. Niemeyer, Tubingen (2004) 314–351
9. Pustejovsky, J.: The Generative Lexicon. MIT Press (1995)
10. Peris, A., Taulé, M.: Evaluaciòn de los criterios linguìsticos para la distinciòn evento y resultado en los sustantivos deverbales. In: Proceedings of the 1st International Conference on Corpus Linguistics (CILC-09). (2009)
11. Pustejovsky, J., Verhagen, M.: SemEval-2010 task 13: Evaluating events, time expressions, and temporal relations (TempEval-2). In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009), Boulder, Colorado, Association for Computational Linguistics (June 2009) 112–116
12. Pianta, E., Bentivogli, L., Girardi, C.: Multiwordnet: Developing and aligned multilingual database. In: Proceedings of the First International Conference on Global WordNet, Mysore, India (January 2002) 293–302
13. Attardi, G., Dell'Orletta, F.: Reverse revision and linear tree combination for dependency parsing. In: NAACL-Short '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, Morristown, NJ, USA, Association for Computational Linguistics (2009) 261–264
14. Hindle, D.: Noun classification from predicate-argument structures. In: Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh, Pennsylvania, USA, Association for Computational Linguistics (June 1990) 268–275
15. Qian, T., Durme, B.V., Schubert, L.: Building a semantic lexicon of english nouns via bootstrapping. In: Proceedings of Human Language Technologies:

The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium, Boulder, Colorado, USA (june 2009)
16. Saurí, R., Knippen, R., Verhagen, M., Pustejovsky, J.: Evita: A robust event recognizer for QA systems. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05). (2005) 700–707
17. Resnik, G., Bel, N.: Automatic detection of non-deverbal event nouns in spanish. In: Proceedings of the 5th International Conference on Generative Approaches to the Lexicon, Pisa: Istituto di Linguistica Computazionale (2009)
18. Eberle, K., Faass, G., Heid, U.: Corpus-based identification and disambiguation of reading indicators for german nominalizations. In: Corpus Linguistics 2009, Liverpool, UK (2009)
19. Peris, A., Taulé, M., Boleda, G., Rodrìguez, H.: Adn-classifier:automatically assigning denotation types to nominalizations. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (May 2010)
20. Creswell, C., Beal, M.J., Chen, J., Cornell, T.L., Nilsson, L., Srihari, R.K.: Automatically extracting nominal mentions of events with a bootstrapped probabilistic classifier. In: COLING-ACL '06: Proceedings of the COLING/ACL on Main conference poster sessions, Morristown, NJ, USA, Association for Computational Linguistics (2006) 168–175

IRENE RUSSO
ISTITUTO DI LINGUISTICA COMPUTAZIONALE,
CNR,
ITALY
E-MAIL: <IRENE.RUSSO@ILC.CNR.IT>

TOMMASO CASELLI
ISTITUTO DI LINGUISTICA COMPUTAZIONALE,
CNR,
ITALY
E-MAIL: <TOMMASO.CASELLI@ILC.CNR.IT>

FRANCESCO RUBINO
ISTITUTO DI LINGUISTICA COMPUTAZIONALE,
CNR,
ITALY
E-MAIL: <FRANCESCO.RUBINO@ILC.CNR.IT>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 105–119
RECEIVED 27/10/10 ACCEPTED 26/11/10 FINAL 15/02/11

Accelerating the Sequence Annotation using Split Annotation with Active Learning

DITTAYA WANVARIE, HIROYA TAKAMURA, AND MANABU OKUMURA

Tokyo Institute of Technology, Japan

ABSTRACT

Active learning succeeds in reducing the size of labeled corpus while maintaining the high accuracy. However, active learning requires several iterations of the tagger training, which will not be practical when the training of an iteration takes long time. In this paper, we propose to simplify the all-entity labeling task by splitting the task into a set of single-entity labeling subtasks. After all entity types are labeled, we merge the data sets into an all-entity corpus and train the final tagger using the merged set. The proposed method achieved the competitive F_1 to the multi-entity learning but required much less computational time on the CoNLL chunking and named entity recognition data sets.

1 INTRODUCTION

Part-of-speech tagging, text chunking, named entity recognition, and a number of tasks in natural language processing are formulated as a sequence labeling task. Sequence labeling is a kind of classification tasks that predicts an output label for each of the corresponding input token in the sequence. Sequence labeling is also a structured labeling task which has a special property that an output label does not depend only on the input sequence, but also on the other output labels. These dependencies among output labels slow the training of the classifier, and are troublesome for an annotator. Although the labeling of a structured output corpus consumes much of human time and effort, a considerably large size of corpus is still necessary in order to train a precise classifier for a task.

Active learning is proposed to reduce the labeling cost in numerous tasks including sequence labeling [6, 11]. The intuition behind it is that only the informative samples are sufficient for the training in order to achieve high accuracy. Hence, it can reduce the annotation cost from the whole corpus to a set of informative samples in the corpus. The intuition of active learning can also be applied to a substructure level, i.e. only some substructures in the whole structure are informative. Tomanek and Hahn proposed to manually label only the informative subsequences, and automatically label the uninformative parts in the sequence [11]. Wanvarie et al. also proposed similar idea to [11], but they re-estimate the labels of uninformative parts in the training without explicitly labeling them [16]. In this paper, we adopt the method in [16] since it can achieve the similar F_1 to the supervised learning while the method in [11] cannot.

In an active learning framework, the informative set of samples are iteratively extracted from the corpus using the tagger trained on the previously labeled data. Therefore, an annotator has to wait for the training to complete before he/she can start labeling for the next iteration. If the training of the tagger takes long time, the framework will be less practical. One of the factors which causes the long training time is the number of the possible output labels. Although the conditional random fields (CRFs) can take the output labels into account in the training, the number of class labels increases and requires long training time. Cohn et al. proposed a CRFs training technique which simplifies the multi-class classification to a set of binary classifications [1]. Their method succeeded in reducing the training time of CRFs, but still required a fully labeled corpus. Here, we adopt their idea to simplify the labeling task into a set of entity labeling tasks in order to reduce the training time. For example, the labeling of a corpus consisting of 4 entity types will be split into 4 labeling subtasks. Each subtask takes only a single entity type into account.

The rest of this paper starts from the description of conditional random fields in Section 3. We summarize the active learning framework adopted in this paper and the proposed split labeling in Section 4. Section 5 contains the comparison between the proposed split labeling and the conventional all-entity labeling. Finally, we summarize the contribution of this paper and discuss the future work in Section 6.

2 Related Work

Obtaining partially labeled data is easy in many situations. For example, we can exploit the keyword link in Wikipedia text for word boundary in-

formation without any human labeling effort. In the domain adaptation task, Tsuboi et al. showed that the training using partially labeled corpus, augmented with the fully labeled source domain, achieved higher accuracy than the training with the source domain [13]. Culotta and McCallum proposed an annotation framework for an information extraction task in [2], which allows the partial annotation of the document.

Tomanek and Hahn proposed a semi-supervised active learning framework for sequence labeling which requires only informative tokens to be manually labeled [11]. Wanvarie et al. also proposed a similar system in [16, 17], but their system does not require any explicit annotation on uninformative tokens. However, all of the systems in [11, 16, 17] employ the multi-entity CRFs training which we will show later in the experiment section that the multi-entity CRFs requires long training time.

Standard L-BFGS optimizer [4] for CRFs is slow due to the full gradient computation, making the active learning impractical if an annotator has a long waiting time between the labeling iterations. In order to accelerate the training of CRFs, several optimization techniques were proposed such as stochastic gradient descent [15]. Apart from the engineering of the optimization itself, Cohn et el. proposed to simplify the multi-class learning task into a set of binary classification subtasks using error-correcting codes [1]. Their proposed method can reduce both of the training time and memory, with a slight decrease in the accuracy. Tsuruoka et al. proposed a sentence selection technique for sparse corpus annotation using active learning [14]. They also succeeded in extracting almost all of the sequences that contain the target entity within a small amount of CPU time. However, they did not employ the partial annotation. Therefore, they have to label the whole corpus in an all-entity labeling task.

3 CONDITIONAL RANDOM FIELDS (CRFS)

Given that there are an input sequence $\mathbf{x} = (x_1, ..., x_T) \in \mathbf{X}$ composed of T tokens and the corresponding output sequence $\mathbf{y} = (y_1, ..., y_T) \in \mathbf{Y}$, the conventional CRFs proposed by [3] model the probability of $\mathbf{y}$ using the following probability:

$$P_\theta(\mathbf{y}|\mathbf{x}) = \frac{e^{\theta \cdot \mathbf{\Phi}(\mathbf{x},\mathbf{y})}}{Z_{\theta,\mathbf{x},\mathbf{Y}}} \,. \tag{1}$$

Z is the normalizing factor:

$$Z_{\theta,\mathbf{x},\mathbf{Y}} = \sum_{\mathbf{y}\in\mathbf{Y}} e^{\theta\cdot\mathbf{\Phi}(\mathbf{x},\mathbf{y})} , \tag{2}$$

which can be computed efficiently using dynamic programming. The feature function Φ is a mapping from $\mathbf{x}$ and $\mathbf{y}$ to a real value. Supposing that we have N training sequences and d features ($\mathbf{\Phi} = (\Phi_1, ..., \Phi_d)$), we will learn the model parameters $\theta = (\theta_1, ..., \theta_d)$, by maximizing the log likelihood of the output sequences given the input sequences in the training data:

$$\max LL(\theta) = \max \sum_{n=1}^{N} ln(P_\theta(\mathbf{y}^{(n)}|\mathbf{x}^{(n)})) . \tag{3}$$

We employ L-BFGS[4] with parallelized gradient computation to optimize θ in (3).

Since the sequence probability in (1) of the objective function in (3) requires a sequence to be fully labeled, we re-define the objective function for partially labeled sequences following [13] to:

$$\max LL(\theta) = \max \sum_{n=1}^{N} \ln P_\theta(\mathbf{Y}_{\mathbf{L}^{(n)}}|\mathbf{x}^{(n)}) . \tag{4}$$

$\mathbf{Y_L}$ is a set of all $\mathbf{y}$ consistent with the partially labeled sequence $\mathbf{x}$. The probability of $\mathbf{Y_L}$ is modified from (1) to

$$P_\theta(\mathbf{Y_L}|\mathbf{x}) = \sum_{\mathbf{y}\in\mathbf{Y_L}} P_\theta(\mathbf{y}|\mathbf{x}) . \tag{5}$$

4 Active Learning Framework

The outline of our labeling framework is shown in Fig.1. The active learning is mostly similar to the framework in [16] augmented with the split labeling. In an iteration of each subtask, the system tries to find a set of informative tokens and ask an annotator to label them. The labeling will stop when the stopping criterion are satisfied. After the annotation of all subtasks has been finished, the partially labeled corpora from all subtasks are merged into a single multi-entity corpus for the training of the final tagger.

$S_{s,t} : \{(\mathbf{x}, \mathbf{y_L})\}$ is a set of all training sequences with current annotation at iteration t of subtask s
S_{sel} is a set of informative tokens
x is an input token
for each s **do**
  $curmodel \leftarrow train(S_{s,1})$ {Initial training}
  repeat
    $S_{sel} \leftarrow Q_{\text{tok}}(curmodel, S_{s,t})$ {Selecting q tokens (at most)}
    for $x \in S_{sel}$ **do**
      $S_{s,t+1} \leftarrow update(S_{s,t}, x, label(x))$ {Annotation}
    end for
    $curmodel \leftarrow train(S_{s,t+1})$ {Training}
  until $(|S_{sel}| < q)$ and $(\kappa(S_{s,t}, S_{s,t+1}) > stop)$ {Stopping criterion}
  $S_{s,final} \leftarrow S_{s,t+1}$
end for
$S_{final} \leftarrow merge(S_{s,final})$ {Merge training sets}
$finalmodel \leftarrow train(S_{final})$

Fig. 1: Active labeling framework

4.1 *Query and Labeling Strategy*

The success key of active learning in reducing the annotation cost relies on the ability to select a small set of highly *informative* labeling candidates. There are various definitions of *informative* candidates such as the prediction confidence or the information gain[6]. In this paper, we define the informativeness of a candidate by its prediction confidence owing to its simplicity. There are also several definitions of the prediction confidence itself. We follow [11, 16, 17] to define the prediction confidence using the marginal probability:

$$P_\theta(y_j = y'|\mathbf{x}) = \frac{\alpha_j(y'|\mathbf{x}) \cdot \beta_j(y'|\mathbf{x})}{Z_\theta(\mathbf{x}, \mathbf{Y})}$$

α and β are the prefix and the suffix probabilities. When the marginal probability of a token is lower than the preset threshold, we regard the token as *informative* and ask an annotator to label the token.

After a token is labeled, the marginal probability of its neighboring tokens will change and may exceeds the threshold. Thus, these tokens will require no labeling effort. The idea is called correction propagation in [2], and probability re-estimation in [16, 17]. We also employ this idea

following [16] by labeling only one token per sequence per iteration, and provide at most q sequences to an annotator in each iteration.

From the objective function in (4), we cannot benefit from the entirely unlabeled sequences. On the contrary, adding unlabeled sequences will slow the optimization process. Therefore, we will train the tagger in a pass using only the fully labeled, and partially labeled sequences.

4.2 *Stopping criterion*

The stopping criterion is also an important key of active learning to reduce the annotation cost by stopping the learning when it converges. Since the labeling is done in token unit, the learning can simply stop after there is no informative token left in the corpus. However, Wanvarie et al. showed in [17] that adding a few new informative tokens does not help improving the F_1 but just wastes the training time.

We employ the stopping criterion in [16], which is described as follows. Firstly, we predict the output of all training sequences, both labeled, partially labeled, and unlabeled using the model in each iteration. Note that the output of a labeled token is exact, and always correct. Then, we measure the similarity between the prediction of the models from two consecutive iterations using Kappa statistic. The learning can be stopped if there are few differences between the prediction. We found that the usual $\kappa = 0.99$ is not sufficient to achieve high accuracy. κ is empirically tuned in the experiments. When the Kappa statistic exceeds the threshold, $\kappa = 0.9999$, the learning will stop.

4.3 *Split Labeling*

The original corpus contains various types of entity tokens. We split the labeling task into a set of single-entity labeling tasks. The labeling in each subtask is also partial labeling. The partially labeled corpora from all subtasks are merged to build the final corpus, which is still partially labeled. The final tagger is trained on the all-entity corpus using CRFs described in section 3.

We assume that there is no ambiguity from human labeling. Therefore, a token which is labeled as an entity type in a subtask will be labeled as a non-entity type in the other subtasks. In contrast, a token labeled as a non-entity type in a subtask may be a real non-entity, or an entity of the other types. A token is regarded as a non-entity token if and only if it is

labeled as a non-entity type in all subtasks. Otherwise, the token is left unlabeled in the merged corpus.

However, a few non-entity tokens are labeled in all subtasks. Therefore, the merged corpus will contain mostly entity tokens, lacking of non-entity tokens, and is not appropriate for the training. We propose to label all of the unlabeled tokens by the model prediction of all subtasks in order to retrieve the non-entity tokens. However, there may be conflicts among the model predictions. In such cases, we leave the tokens unlabeled.

5 Experiments and Discussion

5.1 *Experimental Settings*

Table 1: Data statistics

Data set	#Sequences	#Tokens	#Entity types	#Entity tokens
CoNLL2000				
training	8936	211727	11	183825
test	2012	47377	10	41197
CoNLL2003				
training	14987	204567	4	34043
test	3684	46666	4	8112

We evaluated the proposed method on CoNLL data sets; the chunking task from [9] and the named entity recognition task from [10]. A sentence is represented as a sequence, while a word is represented as a token. The output label for chunking is in IOB2 format [5], while the output label for named entity recognition is in IOB format [8]. Example of each labeling

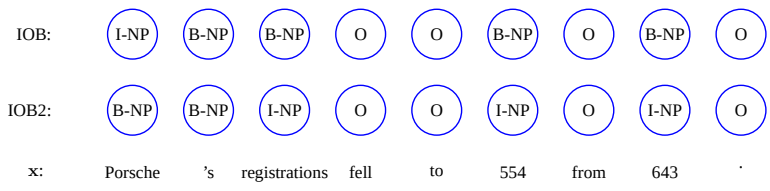


Fig. 2: Labeling example

format is shown in Figure 2. The statistics of each data set is summarized in Table 1. We used the same features described in [16, 17], which are summarized as follows.

The variables w_i and lw_i are the word and its lowercase at i distance from the current word in a sentence. p_i is the part-of-speech (POS) of w_i, which is provided in the data set. c_i is a chunk type which is provided in CoNLL2003 data set, e.g. NP chunk. wtp_i is a set of orthographic features of a word such as containing punctuations. $pw[c]_i$ and $sw[c]_i$ are c-character prefix and suffix of word w_i, e.g. 3-character prefix of the word *American* is *Ame*. y_i is an output label of w_i. Each feature template is shown in a bracket. All templates are augmented with y_i. The subscript and superscript indicate the running index of the word position. For example, $[w_i]_{i=-1}^{i=1}$ refers to three templates, w_{i-1}, w_i, and w_{i+1}.

– CoNLL2000:
  - $[w_i]_{i=-2}^{i=2}$, $[w_i, w_{i+1}]_{i=-1}^{i=0}$, $[p_i]_{i=-2}^{i=2}$, $[p_i, p_{i+1}]_{i=-2}^{i=1}$, $[p_i, p_{i+1}, p_{i+2}]_{i=-2}^{i=0}$, $[y_{i-1}]$
– CoNLL2003:
  - $[w_i]_{i=-1}^{i=1}$, $[w_{i-4}, w_{i-3}, w_{i-2}, w_{i-1}]$, $[w_{i+1}, w_{i+2}, w_{i+3}, w_{i+4}]$
  - $[p_i]_{i=-2}^{i=2}$, $[p_i, p_{i+1}]_{i=-2}^{i=1}$, $[p_{i-1}, p_i, p_{i+1}]$, $[lw_i]_{i=-2}^{i=2}$, $[lw_i, lw_{i+1}]_{i=-2}^{i+1}$, $[wtp_i]_{i=-1}^{i=1}$
  - $[c_i]_{i=-2}^{i=2}$, $[c_i, c_{i+1}]_{i=-2}^{i=1}$, $[c_{i-1}, c_i, c_{i+1}]$
  - $[pw2_i]_{i=-1}^{i=1}$, $[pw3_i]_{i=-1}^{i=1}$, $[sw2_i]_{i=-1}^{i=1}$, $[sw3_i]_{i=-1}^{i=1}$, $[y_{i-1}]$

There are two evaluation criteria; the accuracy which is measured by F_1 using CoNLL evaluation [9], and the annotation cost. The annotation cost is also evaluated in two aspects; the number of manually labeled tokens, and the computational time in seconds.[1] We did not measure the actual annotation time by human annotators but only simulated the human annotation using the gold standard corpus.

The initial training set contains the 47 longest sequences from the training corpus. All tokens in the initial set are manually labeled. The system will provide at most new 500 informative tokens per iteration to an annotator. Although large query size requires more labeled tokens and more annotation time, a few new tokens cannot contribute much to the accuracy improvement but just wastes the computational time. Our objective is to achieve the comparable F_1 to the supervised learning, while

[1] We conducted all experiments on a Xeon 3.0GHz machine. Our CRFs implementation is in C. The optimization of CRFs is parallelized gradient computation of L-BFGS using 4 cpus.

requiring less number of labeled tokens with reasonable training time. Therefore, we should also balance the actual labeling time and the tagger training time. If the tagger training time is approximately 10-15 minutes, the suitable labeling time might be 1-2 hours. If the actual annotation time per token is few seconds [7, 12], the actual labeling time of 500 tokens will be approximately an hour and a half.

5.2 *Baseline Systems*

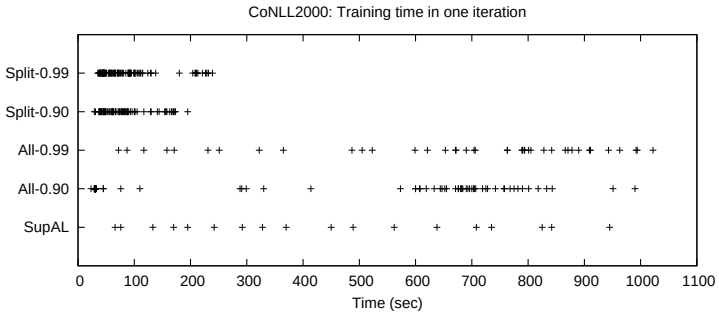
The baselines are all-entity labeling approaches. The first baseline is a supervised active learning system (*SupAL*). In each iteration, an annotator will label all tokens from a set of 500 sequences with the lowest sequence probabilities. In the following experiments, we give a bias to *SupAL* result by reporting the annotation cost when its F_1 reaches the supervised F_1 level. Note that in the real labeling situation, we do not know the real achievement of F_1 in advance. The other two baselines are partial annotation systems using all-entity training (*All*), with the confidence threshold at 0.90, and at 0.99.

5.3 *Result and Discussion*

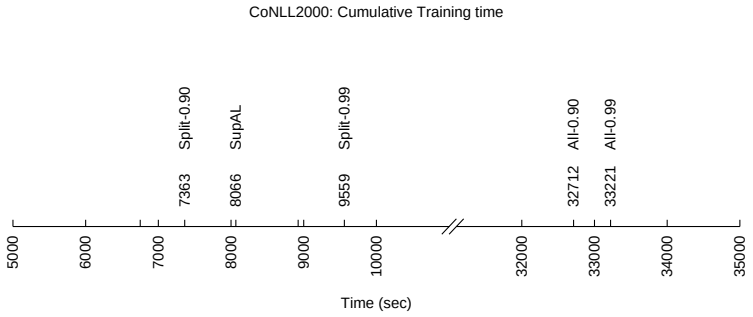
The CPU time per iteration, which is the time an annotator has to wait before start labeling the next iteration, is shown in Figure (3a) and Figure (4a). The CPU time of all settings continued increasing when new labeled tokens were added to the training set. The CPU time in an iteration consists of the time for tagger training, token selection, and evaluating the stopping criterion. Most of the CPU time devotes to the tagger training.

In late iterations of *SupAL* and *All*, an annotator had to wait for more than 10 minutes in CoNLL2000 labeling, and more than 40 minutes in CoNLL2003 labeling. In contrast, the proposed *Split* approach reduced a half of the waiting time in both tasks. There were a few early iterations which require more than an hour in the training of CoNLL2003 experiments, when using the partially labeled training sequences. Note that most of the iterations require less than 40 minutes in the training. We also found similar phenomena in the split labeling but with much smaller amount of training time. We argue that the tagger was uncertain and might incorrectly estimate the output of unlabeled tokens, which produced strange label distribution, resulting in the long training time.

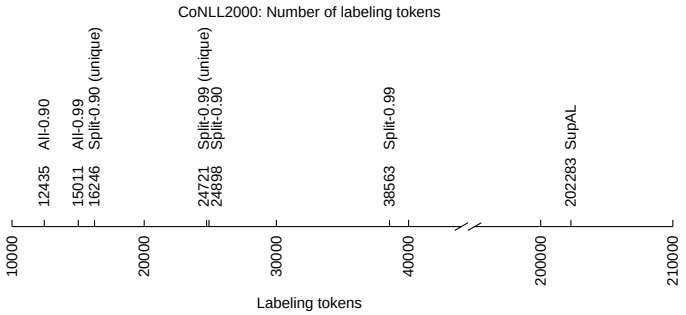
Figure (3b) and Figure (4b) show the cumulative training time of each system. While the split labeling required more training iterations than the all-entity labeling, the cumulative training time was much less.



(a) Computational time of single iteration

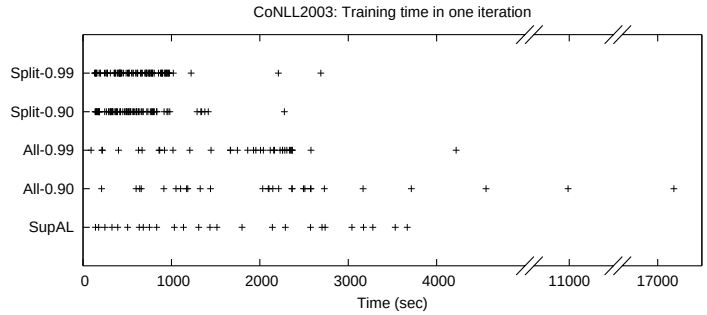


(b) Cumulative computational time

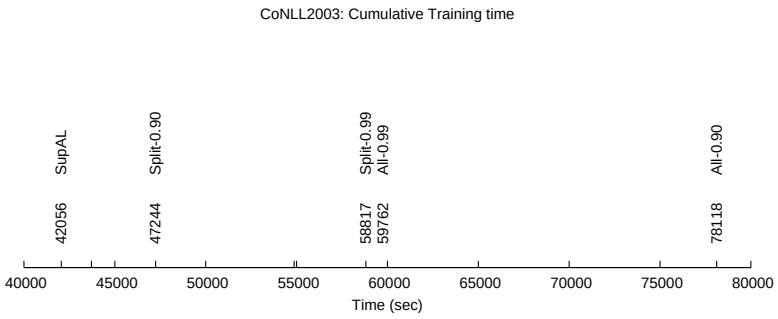


(c) Number of labeling tokens

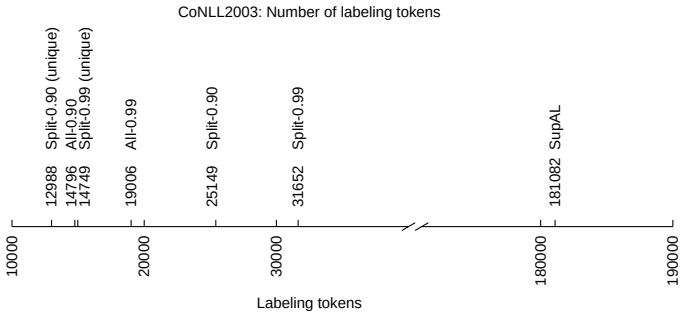
Fig. 3: CoNLL2000



(a) Computational time of single iteration



(b) Cumulative computational time



(c) Number of labeling tokens

Fig. 4: CoNLL2003

Figure (3c) and Figure (4c) show the number of labeled tokens in each data set. We reported the number of unique actions in split labeling using *Split*, and the number of uniquely labeled tokens using *Split (unique)*. With split labeling, a token may be labeled in several subtasks, which results in a number of labeling actions. However, we argue that labeling a single entity type is easy since the number of possible outputs in the candidate list is much less than the number in the all-entity labeling. Moreover, we can parallelize the labeling task by asking a group of annotators to label all subtasks at the same time, which can further accelerate the labeling. Another possible labeling setting is to ask the annotator to find the correct label, but strictly train the model in binary ways.

Table 2: F_1 of each system

Systems	CoNLL2000	CoNLL2003
SupAL	93.42	81.49
All-0.90	93.41	81.21
All-0.99	93.46	81.33
Split-0.90	92.98	80.46
Split-0.99	93.14	81.11

Finally, we compared F_1 of the proposed method with the baselines in Table 2. The proposed method achieved the competitive F_1 to the full labeling settings. The slight reduction of F_1 from the all-entity labeling may due to the lack of inter-entity information since the framework does not distinguish the non-entity and non-target-entity from each other in the sampling of the split subtasks. Another reason is the rare entity samples since the tagger failed to extract sufficient number of such tokens for training.

6 Conclusion and Future Work

In this paper, we have proposed a sequence annotation framework which achieved the competitive accuracy to the supervised system while required much less labeled tokens. The proposed framework also required reasonable training time compared to the exhaustive time of the all-entity labeling framework in the previous work since it could efficiently select the informative tokens in less computational time.

We may be able to further reduce the training time using faster CRFs optimization techniques such as stochastic gradient descent [15], multi-class training [1]. Other learning algorithms such as perceptron, support vector machines are also applicable to our framework. However, we need to re-define the confidence measurement and the optimization for partially labeled sequences.

ACKNOWLEDGMENTS We thank the reviewers for valuable comments and suggestions. The first author is supported from the Thai government scholarship.

REFERENCES

1. Cohn, T., Smith, A., Osborne, M.: Scaling conditional random fields using error-correcting codes. In: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 10–17. Association for Computational Linguistics, Morristown, NJ, USA (2005)
2. Culotta, A., McCallum, A.: Reducing labeling effort for structured prediction tasks. In: AAAI'05: Proceedings of the 20th national conference on Artificial intelligence. pp. 746–751. AAAI Press (2005)
3. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
4. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Math. Program. 45(3), 503–528 (1989)
5. Sang, E.F.T.K., Veenstra, J.: Representing text chunks. In: Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics. pp. 173–179. Association for Computational Linguistics, Morristown, NJ, USA (1999)
6. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1070–1079. Association for Computational Linguistics, Morristown, NJ, USA (2008)
7. Settles, B., Craven, M., Friedland, L.: Active learning with real annotation costs. In: Proceedings of the NIPS Workshop on Cost-Sensitive Learning, 2008 (2008)
8. Tjong Kim Sang, E.F.: Memory-based named entity recognition. In: COLING-02: proceedings of the 6th conference on Natural language learning. pp. 1–4. Association for Computational Linguistics, Morristown, NJ, USA (2002)

9. Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the conll-2000 shared task: chunking. In: Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning. pp. 127–132. Association for Computational Linguistics, Morristown, NJ, USA (2000)
10. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Daelemans, W., Osborne, M. (eds.) Proceedings of CoNLL-2003. pp. 142–147. Edmonton, Canada (2003)
11. Tomanek, K., Hahn, U.: Semi-supervised active learning for sequence labeling. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 1039–1047. Association for Computational Linguistics, Suntec, Singapore (August 2009), http://www.aclweb.org/anthology/P/P09/P09-1117
12. Tomanek, K., Hahn, U., Lohmann, S., Ziegler, J.: A cognitive cost model of annotations based on eye-tracking data. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1158–1167. Association for Computational Linguistics, Uppsala, Sweden (July 2010), http://www.aclweb.org/anthology/P10-1118
13. Tsuboi, Y., Kashima, H., Oda, H., Mori, S., Matsumoto, Y.: Training conditional random fields using incomplete annotations. In: COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics. pp. 897–904. Association for Computational Linguistics, Morristown, NJ, USA (2008)
14. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: Accelerating the annotation of sparse named entities by dynamic sentence selection. In: BioNLP '08: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing. pp. 30–37. Association for Computational Linguistics, Morristown, NJ, USA (2008)
15. Vishwanathan, S.V.N., Schraudolph, N.N., Schmidt, M.W., Murphy, K.P.: Accelerated training of conditional random fields with stochastic gradient methods. In: ICML '06: Proceedings of the 23rd international conference on Machine learning. pp. 969–976. ACM, New York, NY, USA (2006)
16. Wanvarie, D., Takamura, H., Okumura, M.: Active learning with subsequence sampling strategy for sequence labeling tasks., submitted to Journal of Natural Language Processing, Japan
17. Wanvarie, D., Takamura, H., Okumura, M.: Active learning with partially annotated sequence. Tech. rep., Special Interest Group of Natural Language Processing, Information Processing of Japan (September 2010)

DITTAYA WANVARIE
DEPARTMENT OF COMPUTATIONAL INTELLIGENCE AND
SYSTEMS SCIENCE,
TOKYO INSTITUTE OF TECHNOLOGY,
4259 NAGATSUTA-CHO, MIDORI-KU, YOKOHAMA CITY, JAPAN
E-MAIL: <DITTAYA@LR.PI.TITECH.AC.JP>

HIROYA TAKAMURA
PRECISION AND INTELLIGENCE LABORATORY,
TOKYO INSTITUTE OF TECHNOLOGY,
4259 NAGATSUTA-CHO, MIDORI-KU, YOKOHAMA CITY, JAPAN
E-MAIL: <TAKAMURA@PI.TITECH.AC.JP>

MANABU OKUMURA
PRECISION AND INTELLIGENCE LABORATORY,
TOKYO INSTITUTE OF TECHNOLOGY,
4259 NAGATSUTA-CHO, MIDORI-KU, YOKOHAMA CITY, JAPAN
E-MAIL: <OKU@PI.TITECH.AC.JP>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 121–137
RECEIVED 01/11/10 ACCEPTED 26/11/10 FINAL 09/02/11

A Multi-Dimensional Classification Approach towards Recognizing Textual Semantic Relations

RUI WANG AND YI ZHANG

Saarland University and DFKI GmbH, Germany

ABSTRACT

Recognizing textual entailment has been known as a challenging task, with many proposed approaches focusing on solving it independently. From a broader perspective, there are other semantic relations between pairs of texts, e.g., paraphrase, contradiction, overlapping, independence, etc. In this paper, we propose three basic measurements: relatedness, inconsistency, and inequality, to characterize these closely related Textual Semantic Relations. *We show empirically the effectiveness of these measurements for the recognition tasks (e.g. an improvement of 3.1% of accuracy for entailment recognition) with features extracted from dependency paths of the joint syntactic and semantic graph. With the semantic relation space based on these three dimensions, we show this is a way to achieve a better understanding of general semantic relations between texts.*

1 INTRODUCTION

Recognizing Textual Entailment (RTE) has been known as a challenging task, with interesting close relations to both natural language understanding (i.e. meaning interpretation) and natural language processing (i.e. applicable to various tasks). The task was defined as to recognize a specific relation (i.e. *entailment*) between two texts, *text* (T) and *hypothesis* (H). While many attempts have been made to solve the problem in a standalone manner, fewer investigated the relation between entailment and other possible semantic relations between pairs of texts.

From this perspective, most approaches fall into two groups. In the first group, either the system deals with different cases of entailment with specialized modules [1, 2], to learn various lexical or inference rules [3, 4] from large scale corpora, or applies logic inference techniques with manually-crafted rules [5]. In the second group, people work on (seemingly) different tasks, e.g. identifying *contradiction* [6], acquiring paraphrase [7], and finding answers to the questions [8], and try to connect these tasks with the RTE research. This paper falls into this category, too.

The term *semantic relation* refers to the relations that hold between the meaning of two linguistic units. It is commonly used to describe relations between pairs of words, e.g., synonym, hypernym, etc. However, it has also been used in a wider sense to refer to relations between larger linguistic expressions or texts, such as paraphrasing, textual entailment, etc. [9]. We refer to the latter relations as *Textual Semantic Relations* (TSRs), to differentiate them from the study of lexical semantic relations. At a first glance, such generalization makes the already challenging recognition tasks even more complex. However, if these TSRs are mutually related, the simultaneous prediction will make much sense.

In previous work, [10] have shown that recognizing *relatedness* between two texts can be viewed as an intermediate step for entailment and contradiction recognition. [11] proposed five elementary relations between text pairs, EQUIVALENT, FORWARD (ENTAILMENT), REVERSE (ENTAILMENT), INDEPENDENT, and EXCLUSIVE and represent them in terms of entailment and negation. [12] proposed an annotation scheme for semantic relations between text pairs, including six labels, BACKWARD ENTAILMENT, FORWARD ENTAILMENT, EQUALITY, CONTRADICTION, OVERLAPPING, and INDEPENDENT.

In order to obtain a better characterization of all these TSRs, in this paper, we propose three basic numerical features, *relatedness*, *inconsistency*, and *inequality*. We show empirically these features are effective for the TSR recognition tasks, e.g. an improvement of 3.1% of accuracy on entailment recognition and 2.3% on paraphrase identification (Section 5.2). Although these three values are not entirely orthogonal to each other, we can still build an approximate three-dimensional semantic relation space, and observe distributional difference between various TSRs.

2 RELATED WORK

While textual entailment analysis is now widely spotted in many NLP applications, e.g. question answering [13] and machine translation evalua-

tion [14], the state-of-the-art performance of RTE systems is far from satisfactory. According to the yearly RTE challenges (from RTE-1 in 2005 [15] to RTE-5 in 2009 [16]), the average performance of the particpating systems is around 60% on the two-way annotated data (ENTAILMENT vs. NON-ENTAILMENT) and even worse on the three-way annotated data (ENTAILMENT, CONTRADICTION, and UNKNOWN) introduced from the RTE-3 pilot task[1]. Nevertheless, successful systems include both machine-learning-based classifier [17] and logic-form-based inferencer [18].

A variant of the logic inference rule is the textual inference rule or other (syntactic or semantic) representations closer to the surface text than the logic form. The DIRT rule collection [19] has been applied to the RTE task, although the improvement is limited [20]. [4] acquired unary rules instead of the binary DIRT-style rules and showed improvement on the accuracy, although it is still far from satisfactory. Both the logic-rule-based and textual-rule-based systems suffer from either a laborious and fragile system with hand-crafted rules (i.e. being lack of recall) or a large collection of "noisy" rules (i.e. being lack of precision). In order to avoid these disadvantages, we will treat RTE as a classification task and apply feature-based machine learning techniques to achieve robustness.

As for the feature space of the machine learning approaches, tree and graph structures are widely considered. For instance, [21] and their following work used tree editing distance algorithms; and [22] chose a graph matching method. An alternative to the feature engineering attempts, support vector machines (SVMs) with different kernels are also popular in this classification task. Both the (constituent) tree kernel [23, 24] and the subsequence kernel based on syntactic dependency paths [25] were quite successful. Therefore, in our work, we will also use an SVM-based classifier. Instead of using the tree kernels, we extract features based on both syntactic and semantic dependency paths (or triples) as an approximation of the meaning, which greatly reduce the number of dimensions of the feature vectors and achieve better efficiency.

As we mentioned in the introduction, besides RTE, the main goal of this paper is to build a general framework for recognizing different TSRs. Previous work on this aspect includes [11]'s proposal of five elementary relations between texts and our own inventory of six semantic relations [12]. [11] tested their natural logic system on the FraCaS dataset [26], which is manually constructed and focuses more on the different linguistic (semantic) phenomena. While the system achieved quite good results

[1] `http://nlp.stanford.edu/RTE3-pilot/`

on this "text-book" style dataset, the evaluation on the real world texts (e.g. the RTE datasets) did not show much advantage of their approach.

Apart from the entailment recognition, [6] attempted to discover contradiction, although it was then proved to be an even harder problem. There is also rich literature on paraphrase (which can be viewed as a bi-directional entailment relation) acquisition and application [27, etc.]. [9] mainly focused on EQUIVALENCE and CONTRADICTION recognition in terms of subjective texts, i.e. opinions. The recent work by [8] proposed a generic system based on a tree editing model to recognize textual entailment, paraphrase, and answers to questions. We follow this line of research and draw a more general picture of all these semantic relations.

3 TEXTUAL SEMANTIC RELATIONS

We firstly introduce the TSRs we consider in this paper, and then the three features we use to characterize the different relations.

In a previous study [12], we have proposed six relations, BACKWARD ENTAILMENT, FORWARD ENTAILMENT, EQUALITY, CONTRADICTION, OVERLAPPING, and INDEPENDENT. If we consider the unidirectional relations between an ordered pair of texts (i.e. from the first one (**T**) to the second one (**H**)), the first two relations can be collapsed into one. We use the name ENTAILMENT, but we mean a strict directional relation, i.e. **T** entails **H**, but **H** does not entail **T**. The original goal of having both OVERLAPPING and INDEPENDENT is to capture the spectrum of relatedness. However, in practice, even the human annotators found it difficult to agree on many cases. Therefore, we also collapse the last two relations into one, UNKNOWN, following the RTE label convention. After changing EQUALITY into PARAPHRASE, the TSRs we mention in the rest of the paper would be, CONTRADICTION (C), ENTAILMENT (E), PARAPHRASE (P), and UNKNOWN (U).

Although semantic relations are supposed to be situation-independent (i.e. consistently true or false in every possible world), in practice, every text pair is always in a certain context. Our goal here is to differentiate these four TSRs using some latent features shared by them, instead of verifying them in all possible worlds. We assume, there exists a simplified low-dimension semantic relation space. While the identification of effective dimensions is a complex question (see Section 5.3 for more detailed discussion), we start only with three dimensions: *Relatedness (Rel)*, *Inconsistency (Inc)*, and *Inequality (Ine)*, and assume that the different TSRs would be scattered on this space with different distributions.

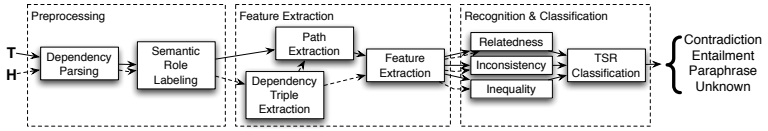


Fig. 1. Workflow of the System

Relatedness captures how relevant the two texts are. PARAPHRASE would be one extreme (fully related), and UNKNOWN would the other extreme. *Inconsistency* measures whether or how contradictory the two texts are. CONTRADICTION has the highest inconsistency, and the others do not have. *Inequality* mainly differentiates the asymmetric ENTAILMENT from the symmetric PARAPHRASE. Although the other two relations are symmetric as well, we assume unequal information is contained in **T** and **H**. All three features will be numerical.

There are two approximations here: i) the number of dimensions in the real semantic relation space is much higher; ii) these three dimensions we pick are not really orthogonal to each other (as shown in the experiments). Nevertheless, we hope to benefit from the generality of these measures in the TSR recognition task and will show the empirical results in Section 5.

4 GENERAL FRAMEWORK

The workflow of the system is shown in Figure 1 and the details of the important components will be elaborated on in the following sections.

4.1 *Preprocessing*

In this paper, we generally refer to all the linguistic analyses on the texts as *preprocessing*. The output of this procedure is a unified graph representation, which approximates the meaning of the input text. In particular, after tokenization and POS tagging, we did dependency parsing and semantic role labeling.

Tokenization and POS Tagging We use the Penn Treebank style tokenization throughout the various processing stages. **TnT**, an HMM-based POS tagger trained with Wall Street Journal sections of the PTB, was used to automatically predict the part-of-speech of each token in the texts and hypotheses.

Dependency Parsing For obtaining the syntactic dependencies, we use two dependency parsers, MSTParser [28] and MaltParser [29]. MSTParser is a graph-based dependency parser where the best parse tree is acquired by searching for a spanning tree which maximize the score on an either partially or fully connected dependency graph. MaltParser is a transition-based incremental dependency parser, which is language-independent and data-driven. It contains a deterministic algorithm, which can be viewed as a variant of the basic shift-reduce algorithm. The combination of two parsers achieves state-of-the-art performance.

Semantic Role Labeling The statistical dependency parsers provide shallow syntactic analyses of the entailment pairs through the limited vocabulary of the dependency relations. In our case, the CoNLL shared task dataset from 2008 were used to train the statistical dependency parsing models. While such dependencies capture interesting syntactic relations, when compared to the parsing systems with deeper representations, the contained information is not as detailed. To compensate for this, we used a shallow semantic parser to predict the semantic role relations in the **T** and **H** of entailment pairs. The shallow semantic parser was also trained with CoNLL 2008 shared task dataset, with semantic roles extracted from the Propbank and Nombank annotations [30].

4.2 *Feature Extraction*

We firstly extract all the dependency triples from **H**, like <word, dependency relation, word>, excluding those having stop words. Then, we use the word pairs contained in the extracted dependency triples as anchors to find the corresponding *dependency paths* in **T**. For the following three representations, we apply slightly different algorithms to find the dependency path between two words,

Syntactic Dependency Tree We traverse the tree to find the corresponding dependency path connecting the two words;

Semantic Dependency Graph We use Dijkstra's algorithm to find the shortest path between the two words;

Joint Dependency Graph We assign different weights to syntactic and semantic dependencies and apply Dijkstra's algorithm to find the shortest path (with the lowest cost)[2].

[2] In practice, we simply set semantic dependency costs at 0.5 and syntactic dependency costs at 1.0, to show the preferences on the former when both exist.

For the features, we firstly check whether there are dependency triples extracted from **H** as well as whether the same words can be found in **T**. Only if the corresponding dependency paths are successfully located in **T**, we could extract the following features. The direction of each dependency relation or path could be interesting. We use a boolean value to represent whether **T**-path contains dependency relations with different directions of the **H**-path.

Notice that all the dependency paths from **H** have length 1[3]. If the length of the **T**-path is also 1, we can directly compare the two dependency relations; otherwise, we compare each of the dependency relation contained the **T**-path with **H**-path one by one[4]. By comparing the **T**-path with **H**-path, we mainly focus on two values, the category of the dependency relation (e.g. syntactic dependency vs. semantic dependency) and the content of the dependency relation (e.g. A1 vs. AM-LOC). We also incorporate the string value of the dependency relation pair and make it boolean depending on whether it occurs or not.

Table 1. Feature types of different settings of the system.

	H_NULL?	*T_NULL?*	*DIR*	*MULTI?*	*DEP_SAME?*	*REL_SIM?*	*REL_SAME?*	*REL_PAIR*
Syn Dep		+	+	+			+	+
Sem Dep	+	+	+	+		+	+	+
Joint	+	+	+	+	+	+	+	+

Table 1 shows the feature types we extract from each **T**-**H** pair. There, H_NULL? means whether **H** has dependencies; T_NULL? means whether **T** has the corresponding paths (using the same word pairs found in **H**); DIR is whether the direction of the path **T** the same as **H**; MULTI? adds a prefix, M_, to the REL_PAIR features, if the **T**-path is longer than one dependency relation; DEP_SAME? checks whether the two dependency types are the same, i.e. syntactic and semantic dependencies; REL_SIM?

[3] The length of one dependency path is defined as the number of dependency relations contained in the path.

[4] Enlightened by [25], we exclude some dependency relations like “CONJ”, “COORD”, “APPO”, etc., heuristically, since usually they will not change the relationship between the two words at both ends of the path.

only occurs when two semantic dependencies are compared, meaning whether they have the same prefixes, e.g. C-, AM-, etc.; REL_SAME? checks whether the two dependency relations are the same; and REL_PAIR simple concatenates the two relation labels together.

4.3 *TSR Recognition*

After obtaining all the features for text pairs with different TSRs, we train three classifiers for the three measurements, *relatedness*, *inconsistency*, and *inequality*, and test on the whole dataset to obtain the numerical values. The training data are labeled according the scheme shown in Table 2. The later recognition of the TSRs are based on these three measurements.

Table 2. Training data of the three classifiers

	relatedness	*inconsistency*	*inequality*
PARAPHRASE	+	−	−
ENTAILMENT	+	−	+
CONTRADICTION	+	+	+
UNKNOWN	−	−	+

5 EXPERIMENTS

5.1 *Datasets*

Table 3 gives an overview of all the datasets we use in our experiments and we briefly describe them in the following.

AMT is a dataset we constructed using the crowd-sourcing technique [31]. We used Amazon's Mechanical Turk[5], online non-expert annotators [32] to perform the task. Basically, we show the Turkers a paragraph of text with one highlighted named-entity and ask them to write some facts or counter-facts about it. There are three blank lines given for the annotators to fill in. For each task, we show five texts, and for each text, we ask three Turkers to do it. In all, we collected 406 valid facts and 178 counter-facts, which will be viewed as E and C respectively.

MSR is a paraphrase corpus provided by Microsoft Research [33]. It is a collection of manually annotated sentential paraphrases. This dataset

[5] `https://www.mturk.com/mturk/`

Table 3. Collection of heterogenous datasets with different annotation schemes.

Corpora	Paraphrase (P)	Entailment (E)	Contradiction (C)	Unknown (U)
AMT (584)		Facts (406)	Counter-Facts (178)	
MSR (5841)	Paraphrase (3940)	Non-Paraphrase (1901)		
PETE (367)		YES (194)	NO (173)	
RTE (2200)	ENTAILMENT (1100)		CONTRADIC-TION (330)	UNKNOWN (770)
TSR (260)	Equality Entailment (3)	Forward/Back-ward (10/27)	Contradiction (17)	Overlapping & Independent (203)
Total (9252)	3943	637	525	973

consists of 5841 pairs of sentences which have been extracted from news sources on the web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship.

PETE is taken from the SemEval-2010 Task #12, Parser Evaluation using Textual Entailment[6] [34]. The dataset contains 367 pairs of texts in all and has a focus on entailments involving mainly the syntactic information. The annotation is two-way, YES would be converted into ENTAILMENT and NO could be either CONTRADICTION or UNKNOWN. Since each text pair only concerns about one syntactic phenomenon, the entailment relation is directional, excluding the paraphrases.

RTE is a mixture of RTE-4 (1000) and RTE-5 (1200) datasets. Both are annotated in three-way, but the ENTAILMENT cases actually include PARAPHRASE as well. We did not include the unofficial three-way annotation of the RTE-3 pilot task.

TSR is the dataset we annotated under the annotation scheme mentioned in Section 3. The sentence pairs were extracted from the the RST Discourse Treebank (RST-DT)[7]. The annotation was done by two annotators in two rounds. The inter-annotator agreement is 91.2% and the kappa score is 0.775. We take all the valid and agreed sentence pairs (260) as the TSR dataset here.

[6] `http://pete.yuret.com/guide`

[7] Available from the LDC: `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07`

We randomly sample 250 **T**-**H** pairs from each dataset as the test sets (1000 pairs in all). The rest of the data are then randomly selected to create a balance training set with equal number of instance pairs from each class.

5.2 *Setup & Results*

First, we take the PETE dataset to do binary classification (ENTAILMENT vs. NON-ENTAILMENT) on a small scale to confirm that both syntactic and semantic dependency structures are useful. The features extracted from the joint dependency graph improve the model of features purely from the syntactic dependency tree by as much as 10% of accuracy. Therefore, in the rest of the experiments, we will take the joint dependency graph as the default structure to extract features.

For comparison, we configure our system in the following two ways to compose different baseline systems: 1) from the classification strategy perspective, the direct four-class classification would be the baseline (*Direct Joint* in Table 4), compared with the main system with a two-stage classification (*3-D Model*); and 2) from the feature set point of view, we take the bag-of-words similarity as the baseline[8] (*Direct BoW*), compared with the main system using both syntactic and semantic dependency structures (i.e. the *3-D Model*). Table 4 shows the results.

Table 4. Results of the system with different configurations and different evaluation metrics.

Systems	4-Way	3-Way	2-Way	
	(C, E, P, U)	(C, E&P, U)	(E&P, Others)	(P, Others)
Direct BoW	39.3%	54.5%	63.2%	62.1%
Direct Joint	42.3%	50.9%	66.8%	77.3%
3-D Model	**45.9%**	**58.2%**	**69.9%**	**79.6%**

Notice that E here indicates the strict directional entailment excluding the bidirectional ones (i.e. P), which makes the task much harder (as we will see it more in Section 5.3). Nevertheless, the main approach, 3-D Model, improves the system performance greatly in all aspects, compared with the baselines. Apart from the self-evaluation, we also compare

[8] The bag-of-words similarity has shown to be a strong baseline in the previous RTE challenges.

our approach with others' systems. Due to the difference in datasets, the numbers are only indicative.

Table 5. System comparison under the RTE annotation schemes (∗ indicates different datasets).

RTE	3-Way	2-Way		
	(C, E&P, U)	Acc.	Prec.	Rec.
3-D Model	58.2%	**69.9%**	**75.9%**	53.4%
M&M, 2007(NL)	–	59.4%	70.1%	36.1%
H&S, 2010	–	62.8%	61.9%	**71.2%**
Our Prev.	**59.1%**	69.2%	–	–
RTE-4 Median	50.7%	61.6%	–	–
RTE-5 Avg.	52.0%	61.2%	–	–

For the RTE comparison (Table 5), the datasets are partially different due to the mixture of datasets. For reference, we re-run our previous system on the new dataset (indicated as *Our Prev.*, which was one of the top system in the previous RTE challenges). The results show that our new approach (*3-D Model*) catches the previous system on the three-way RTE and outperforms it on the two-way task. And both systems achieves much better results than the average. [11]'s system based on natural logic (*M&M, 2007*) is precision-oriented while [8]'s (*H&S, 2010*) is recall-oriented. Our system achieves the highest precision among them.

Table 6. System comparison under the paraphrase identification task (∗ indicates the test sets).

P vs. Non-P	Acc.	Prec	Rec.
3-D Model	**79.6%**	57.2%	72.8%
D&S, 2009 (QG)	73.9%	74.9%	**91.3%**
D&S, 2009 (PoE)	76.1%	**79.6%**	86%
H&S, 2010	73.2%	75.7%	87.8%

Besides the RTE task, we also compare our approach with other paraphrase identification systems (Table 6). [35] proposed two systems, one with high-recall (*D&S, 2009 (QG)*, using a quasi-synchronous grammar) and the other with high-precision (*D&S, 2009 (PoE)*, using a product of experts to combine the QG model with lexical overlap features). *H&S,*

2010 is the same system in Table 5. Although our system has lower precision and recall, our accuracy ranks the top, which indicates that our approach is better at non-paraphrase recognition.

Notice that, our system is not fine-tuned to any specific recognition task. Instead, we build a general framework for recognizing all the four TSRs. We also include heterogenous datasets collected by various methods in order to achieve the robustness of the system. On the contrary, if one is interested in recognizing one specific relation, a closer look at the data distribution would help with the feature selection.

5.3 *Discussion*

While the empirical results show a practical advantage of applying the three-dimensional space model in the TSR recognition task, in this subsection, we investigate whether this simplified semantic relation space with the chosen axises is a good approximation for these TSRs. We plot all the test data into this space and Figure 2 shows three different projections onto each two-dimensional plane.

Although the improvement on recognition accuracy is encouraging, these three measurements cannot fully separate different TSRs in this space. P is clearly differentiated from the others and most of the data points stay in the region of low inconsistency (i.e. consistent), low inequality (i.e. equal), and high relatedness. However, the other three TSRs behave rather similarly to each other in terms of the regions.

Figure 3 shows the other three TSRs on the same plane, *inconsistency-inequality*. Although the general trend of these three groups of data points is similar, slight differences do exist. U is rather restricted in the region of high inconsistency and high inequality; while the other two spread a bit over the whole plane. We have expected the contrary behavior of C and E in terms of inconsistency, but it seems that our inconsistency measuring module is not as solid as the relatedness measure. This is in accordance with the fact that for the original three-way RTE task C is also the most difficult category to be recognized.

A even more difficult measurement is the inequality. Among all the four TSRs, the worst result is on E, which roots from the suboptimal inequality recognition. In retrospect, the matching methods applied to the **T**-**H** pair cannot capture the directionality or the semantic implication, but rather obtain a symmetric measurement, and thus it explains the success of paraphrase recognition. Additionally, this might also suggest that, in the traditional RTE task, the high performance might attribute to the

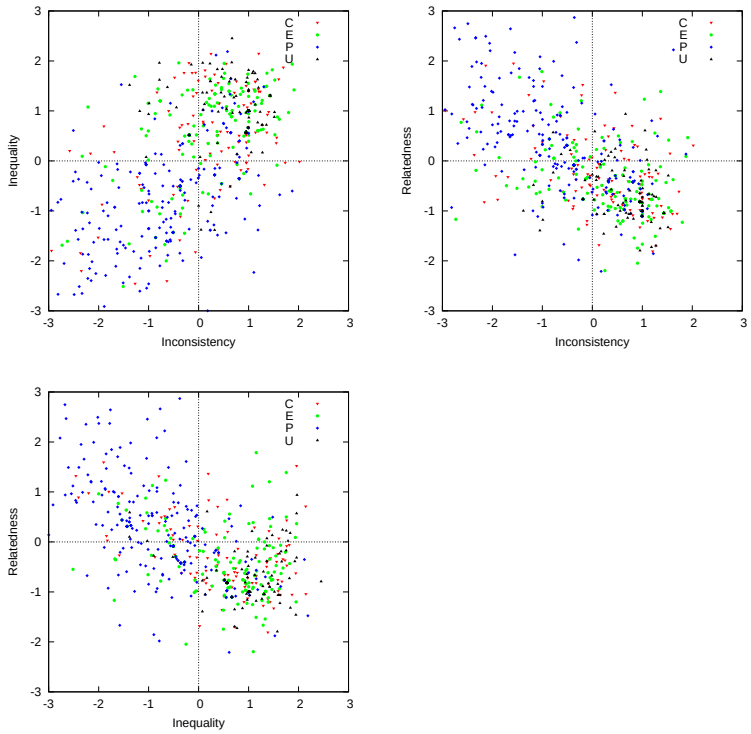


Fig. 2. Test data in the three-dimensional semantic relation space projected onto the three planes.

P "section" of the entailment, while the real directional E is still very difficult to capture.

6 Conclusion and Future Work

In this paper, we present our approach of recognizing different textual semantic relations based on a three-dimensional model. *Relatedness*, *inconsistency*, and *inequality* are considered as the basic measurements for the recognition task as well as the dimensions of the semantic relation space. We show empirically the effectiveness of this approach with a feature model based on dependency paths of the joint syntactic and semantic graph. We also interpret the results and the remaining difficulties visually.

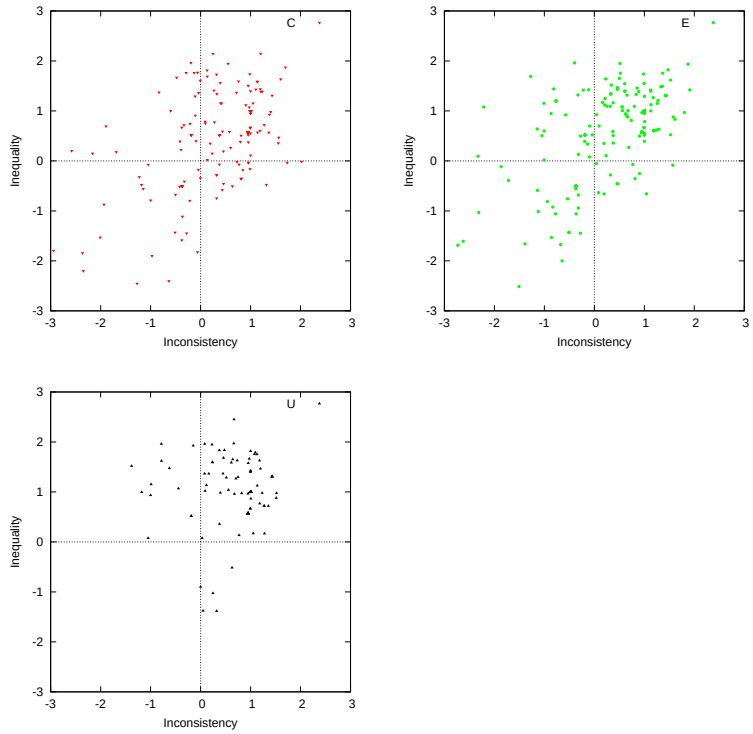


Fig. 3. C, E, and U test data projected onto the inconsistency-inequality plane.

There are several issues on the list: 1) Inequality seems to be difficult to define and measure, which suggests to consider other possible dimensions; 2) we are looking for a systematic way to tune the general system for specific TSR recognition tasks; and 3) we have not incorporated lexical resources (e.g. WordNet) into our system yet, for a proper way of integration is still up for future research.

ACKNOWLEDGMENT The first author is funded by the PIRE PhD scholarship program and the EuroMatrixPlus project (IST-231720) which is funded by the European Commission under the Seventh Framework Programme. The second author thanks the German Excellence Cluster of Multimodal Computing and Interaction for the support of the work. Many

thanks to Hans Uszkoreit for the useful discussion and Sebastian Riedel for the tool to visualize dependency trees/graphs.

REFERENCES

1. Wang, R., Neumann, G.: An accuracy-oriented divide-and-conquer strategy for recognizing textual entailment. In: Proceedings of TAC Workshop. (2009)
2. Cabrio, E.: Specialized entailment engines: Approaching linguistic aspects of textual entailment. In: Natural Language Processing and Information Systems. Springer (2010) 305–308
3. Szpektor, I., Shnarch, E., Dagan, I.: Instance-based evaluation of entailment rule acquisition. In: Proceedings of ACL. (2007) 456–463
4. Szpektor, I., Dagan, I.: Learning entailment rules for unary templates. In: Proceedings of COLING 2008, Manchester, UK (2008) 849–856
5. Bos, J., Markert, K.: Recognising textual entailment with logical inference. In: Proceedings of HLT-EMNLP 2005. (2005) 628–635
6. de Marneffe, M.C., Rafferty, A.N., Manning, C.D.: Finding contradictions in text. In: Proceedings of ACL-08: HLT. (2008)
7. Bar-Haim, R., Berant, J., Dagan, I.: A compact forest for scalable inference over entailment and paraphrase rules. In: Proceedings of EMNLP. (2009)
8. Heilman, M., Smith, N.A.: Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In: Proceedings of NAACL-HLT. (2010) 1011–1019
9. Murakami, K., Masuda, S., Matsuyoshi, S., Nichols, E., Inui, K., Matsumoto, Y.: Annotating semantic relations combining facts and opinions. In: ACL-IJCNLP '09: Proceedings of the Third Linguistic Annotation Workshop. (2009)
10. Wang, R., Zhang, Y.: Recognizing textual relatedness with predicate-argument structures. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), Singapore, Singapore, Association for Computational Linguistics (2009)
11. MacCartney, B., Manning, C.D.: Natural logic for textual inference. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. (2007) 193–200
12. Wang, R., Sporleder, C.: Constructing a textual semantic relation corpus using a discourse treebank. In: Proceedings of the seventh international conference on Language Resources and Evaluation (LREC), Valletta, Malta (2010)
13. Harabagiu, S., Hickl, A., Lacatusu, F.: Negation, contrast and contradiction in text processing. In: Proceedings of AAAI. (2006)
14. Padó, S., Cer, D., Galley, M., Jurafsky, D., Manning, C.D.: Measuring machine translation quality as semantic equivalence: A metric based on entailment features. Machine Translation **23**(2–3) (2009) 181–193

15. Dagan, I., Glickman, O., Magnini, B.: The pascal recognising textual entailment challenge. In: Machine Learning Challenges. Lecture Notes in Computer Science. Springer (2006)
16. Bentivogli, L., Magnini, B., Dagan, I., Dang, H., Giampiccolo, D.: The fifth pascal recognizing textual entailment challenge. In: Proceedings of the Text Analysis Conference (TAC 2009) Workshop. (2009)
17. Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B., Shi, Y.: Recognizing textual entailment with LCC's groundhog system. In: Proceedings of the RTE-2 Workshop. (2006)
18. Tatu, M., Moldovan, D.: A logic-based semantic approach to recognizing textual entailment. In: Proceedings of the COLING/ACL on Main conference poster sessions. (2006) 819–826
19. Lin, D., Pantel, P.: Dirt - discovery of inference rules from text. In: In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. (2001) 323–328
20. Dinu, G., Wang, R.: Inference rules and their application to recognizing textual entailment. In: Proceedings of EACL. (2009) 211–219
21. Kouylekov, M., Magnini, B.: Recognizing textual entailment with tree edit distance algorithms. In: Proceedings of the PASCAL Challenges on RTE. (2005) 17–20
22. Haghighi, A., Ng, A., Manning, C.: Robust textual inference via graph matching. In: Proceedings of HLT-EMNLP 2005. (2005) 387–394
23. Zanzotto, F.M., Moschitti, A.: Automatic learning of textual entailments with cross-pair similarities. In: Proceedings of the COLING/ACL, Sydney, Australia (2006) 401–408
24. Mehdad, Y., Moschitti, A., Zanzotto, F.M.: Syntactic/semantic structures for textual entailment recognition. In: Proceedings of NAACL-HLT. (2010) 1020–1028
25. Wang, R., Neumann, G.: Recognizing textual entailment using a subsequence kernel method. In: Proceedings of AAAI. (2007) 937–942
26. Cooper, R., Crouch, D., Eijck, J.V., Fox, C., Genabith, J.V., Jaspars, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M., Pulman, S.: A framework for computational semantics (FraCaS). Technical report, The FraCaS Consortium (1996)
27. Barzilay, R., Lee, L.: Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In: Proceedings of NAACL-HLT. (2003)
28. McDonald, R., Pereira, F., Ribarov, K., Hajic, J.: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In: Proceedings of HLT-EMNLP 2005. (2005) 523–530
29. Nivre, J., Nilsson, J., Hall, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering **13**(1) (2007) 1–41
30. Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., Nivre, J.: The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In: Proceedings of CoNLL-2008. (2008)

31. Wang, R., Callison-Burch, C.: Cheap facts and counter-facts. In: Proceedings of NAACL-HLT 2010 Workshop on Amazon Mechanical Turk, Los Angeles, California (2010)
32. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks. In: Proceedings of EMNLP. (2008)
33. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Proceedings of the IWP2005. (2005)
34. Yuret, D., Han, A., Turgut, Z.: Semeval-2010 task 12: Parser evaluation using textual entailments. In: Proceedings of the SemEval-2010 Evaluation Exercises on Semantic Evaluation. (2010)
35. Das, D., Smith, N.A.: Paraphrase identification as probabilistic quasi-synchronous recognition. In: Proceedings of ACL-IJCNLP 2009. (2009)

RUI WANG
SAARLAND UNIVERSITY AND DFKI GMBH
SAARBRUECKEN, 66123,
GERMANY
E-MAIL: <RWANG@COLI.UNI-SB.DE>

YI ZHANG
SAARLAND UNIVERSITY AND DFKI GMBH
SAARBRUECKEN, 66123,
GERMANY
E-MAIL: <YZHANG@COLI.UNI-SB.DE>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 139–154
RECEIVED 01/11/10 ACCEPTED 26/11/10 FINAL 08/02/11

In Search of the Use-Mention Distinction and its Impact on Language Processing Tasks

SHOMIR WILSON

University of Maryland at College Park, USA

ABSTRACT

The use-mention distinction is a crucial aspect of natural language which allows us to communicate information about language itself. However, the distinction remains underexamined in computational linguistics, with deleterious effects on common tasks. One reason for this deficiency is a lack of appropriate resources to study the distinction. This paper presents the creation of a corpus of instances of mentioned language gathered from Wikipedia, using a set of "mention words" and cues in text formatting. The corpus demonstrates that recurring patterns do exist among instances of mentioned language, which suggests the potential for automatic identification of the phenomenon in the future.

KEY WORDS: metalanguage, corpus, parsing

1 INTRODUCTION

In order to understand the language that we speak, we sometimes must refer to the language itself. Language users are able to do this through an awareness of the use-mention distinction, which separates language produced to illustrate properties of itself from language produced for other reasons. This can be demonstrated in a pair of simple sentences:

The cat is on the mat. (1)
The word cat refers to a feline animal. (2)

A reader easily understands that *cat* in the first sentence refers to an animal entity in a real or hypothetical world, while the same word in the second sentence refers to the word *cat* itself. The use-mention distinction is well-known and has a history of theoretical examination [1-4], but its actual patterns of appearance in natural language have received little study. This lack of attention has a deleterious impact on common tasks in computational linguistics and natural language processing. Part-of-speech taggers assume by their nature that words are used—which is quite reasonable, since the vast majority of language production *is* use—but this means serious issues arise when language is mentioned. While mentioned words (such as "cat" in (2) above) ostensibly serve as nouns or noun phrases, words that rarely (or never) appear as nouns otherwise are subject to mention as well. For similar reasons, mentioned language also interferes with word sense disambiguation; senses imply language use but have little relevance when a word appears simply "as a word". Mention also has the potential to interfere with sentiment analysis as well; one can discuss another's disapproval of something, for instance, without actually disapproving of anything.

The historical lack of attention to the use-mention distinction might suggest that it is peripheral to the study of language, but this is far from the truth. Evidence suggests that human communication frequently employs the use-mention distinction, and we would be severely handicapped without it [5, 6]. In both written and spoken contexts, the mention of letters, sounds, words, phrases, or entire sentences is essential for many language activities, including the introduction of new words, attribution of statements, explanation of meaning, and assignment of names [7]. The distinction is also closely related to the appearance-reality distinction in cognitive science [8].

Moreover, detecting the distinction is a nontrivial task. While cues like italic text are sometimes used to indicate mentioned language, such cues are often inconsistently applied (if at all, in informal contexts) and are "overloaded" with other uses as well. Cues such as pauses and gestures exist for mentioned language in spoken conversation, but these are easily lost in transcription.

Given the common reasons for employing the use-mention distinction, a text whose purpose is to introduce to the reader a broad swath of concepts would seem a good place to begin studying the

phenomenon. Wikipedia is such a compendium, and several other aspects make it particularly attractive for study. Some of these are its collaborative nature, its stylistic cues (such as italics) to highlight mentioned language, and its size and article variety. Preliminary studies [9] have validated these observations, but they have left open the question of whether instances of mentioned language can be gathered from Wikipedia accurately and in large numbers.

This paper presents the results of a project to identify instances of mentioned language from English Wikipedia articles using lexical and stylistic cues. First, the use-mention distinction is introduced in greater detail, with some examples to illustrate the variety of the phenomenon. A corpus of mentioned language, named the ML corpus for brevity, is then presented, accompanied by a discussion of the lexical and stylistic cues that were used to gather candidates for inclusion. Although multiple human annotators were available for only a subset of the corpus, their frequency of agreement is a mild indication of reliability and consistency. It is believed that this corpus will be sufficient to bootstrap the first efforts for automatic detection of mentioned language.

2 THE USE-MENTION DISTINCTION

Although the reader is likely to be familiar with the use-mention distinction, the topic merits further explanation to establish what precisely is being studied. Since the vast majority of language is produced for use rather than mention, this paper will focus on occurrences of mentioned language. Linguistic entities that can be mentioned include letters, sounds, words, names, phrases, and entire sentences.

2.1 AN INFORMAL RUBRIC

In spite of the ubiquity of the phrase *use-mention distinction,* it is difficult to find any previous efforts to identify when mention does (and does not) occur. For that purpose, this paper will propose an informal rubric based on substitution. It may be applied, with caveats described below, to determine whether a linguistic entity is mentioned by the sentence in which it occurs.

Rubric: Suppose X is linguistic entity in a sentence. X is an instance of mentioned language if the meaning of the sentence does not change when X is replaced by "that [item]*", where* [item] *is "letter", "sound", "word", "name", "phrase", "sentence", etc., and the replacement phrase is understood to refer to X.*

For example, consider the sentence

Fancy automobiles are called luxury cars. (3)

where the phrase "luxury cars" is under consideration. Choosing "that phrase" as a replacement, the sentence becomes

Fancy automobiles are called that phrase. (4)

where "that phrase" is understood to refer to "luxury cars". While there might be pragmatic consequences to this change (for instance, a context where a language user wants to avoid producing the phrase "luxury cars"), the reader can verify that the meaning of the sentence is essentially unchanged.

This rubric requires some adjustment when the sentence already explicitly refers to *X* as a word, phrase, or other appropriate entity, such as in (2) above. In such cases it may be appropriate to omit the linguistic entity under consideration without substituting, such as this alteration to (2):

The word refers to a feline animal. (5)

where "The word" is understood to refer to "cat". Instances of mixed quotation, which straddle both use and mention [10], also may require some charitable adjustments. This is especially apparent when explicit cues of mention are present. An example of this is

Jane said the cat "is on the mat". (6)

where the reader should understand[1]

Jane said the cat "is on the mat", in that exact phrase. (7)

While other permutations exist that challenge the letter of its rubric, this paper will posit that its spirit is sufficiently sound.

[1] Depending on context, instead this might be a use of *scare quotes* to imply that the cat is not actually on the mat. We will assume that was not the intent, although it has been argued [4] that scare quotes are a form of mention.

2.2 CATEGORIES OF MENTIONED LANGUAGE

A previous study by Wilson [9] gathered a small 171-sentence corpus of mentioned language from Wikipedia, in order to demonstrate its fertility as a source and to determine some categories for the phenomenon. The study used a set of eight categories of mentioned language inspired by previous theoretical work [7, 10], with the acknowledgement that others may exist. The categories are reproduced below to illustrate the diversity of forms of mentioned language, with examples from the corpus for each. The mentioned entity of interest in each sentence appears between asterisks, and longer sentences have been shortened with ellipses.

Proper name: In 2005, Ashley Page created another short piece on Scottish Ballet, a strikingly modern piece called *The Pump Room*, set to pulsating music by Aphex Twin.

Translation or transliteration: The Latin title translates as *a method for finding curved lines enjoying properties of maximum or minimum, or solution of isoperimetric problems in the broadest accepted sense*.

Attributed language[2]: *It is still fresh in my memory that I read a chess book of Karpov by chance in 1985 which I liked very much*, the 21-year-old said.

Words or phrases as themselves: *Submerged forest* is a term used to describe the remains of trees (especially tree stumps) which have been submerged by marine transgression…
Symbols: He also introduced the modern notation for the trigonometric functions, the letter *e* for the base of the natural logarithm…

Phonetics or sounds: The call of this species is a high pitched *ke-ke-ke*…

Abbreviations: …Moskovskiy gosudarstvennyy universitet putej soobshcheniya, often abbreviated *MIIT* for “Moscow Institute of Transport Engineers”…

[2] In discussions with other researchers, the author has noted some controversy regarding the inclusion of *attributed language*. While it lacks the "pure mention" quality of some of the other categories, it is discussed as mention (albeit in a “mixed” form) in the cited literature, and the authors would argue that it satisfies the rubric set in 2.1. In the absence of a strong justification for excluding attributed language, this study takes an inclusive approach to it.

Proper names were the most common category found by this previous study, which followed the intuition that many Wikipedia articles describe entities identified by proper names. Translation or transliteration, attributed language, and words or phrases as themselves were the next most common, with fewer instances of the remaining categories.

3 CORPUS CREATION

As explained in the Introduction, a great deal of theoretical study exists on the use-mention distinction, but little (if any) previous research has been concerned with how language users actually exhibit the distinction. The ML Corpus, whose creation is described in this section, is the first substantial attempt to rectify this gap in research. Although it has some limitations, the value of such a resource is not expected to be diminished by them.

3.1 RATIONALE FOR CHOOSING WIKIPEDIA

Wikipedia is a particularly suitable source for collecting instances of mentioned language. Listed here are four factors that led to its selection for this project:

1) *Wikipedia introduces a wide variety of concepts to the reader*. At the time of writing this paper, Wikipedia contains approximately 3.3 million articles. These articles are written informatively, generally without assuming that the reader is already familiar with the topics they discuss. New names and words are frequently introduced, often explicitly, in a manner that invokes mention.

2) *Stylistic cues that are sometimes used to delimit mentioned language are present in article text*. Wikipedia contributors often (though not always) use quote marks, italic text, or bold text to "highlight" where language is mentioned. This convention is stated in Wikipedia's own style manual, though it is unclear whether most contributors read it there or follow it out of habit.

3) *Wikipedia is collaboratively written*. Its text reflects the language habits of a large sample of English writers. It is unclear how much

variation exists between writers on how to mention language, so this large sample is desirable.

4) *Wikipedia is freely available*. Language-learning materials (particularly textbooks) were also considered, but legal issues and electronic availability were deemed obstacles. Moreover, the markup code for Wikipedia articles is easy to access and interpret. This allows for the automatic extraction of the stylistic cues mentioned above.

Naturally, choosing Wikipedia for this project introduced some limitations as well. Since articles are not consistently edited, some mentioned language was not captured by stylistic cues. Such cues are also used by Wikipedia contributors for other purposes, such as emphasis, algebraic symbols, and implicit "non-mention" introduction of words. The particular style of writing in Wikipedia differs from other styles where analysis of mentioned language could be valuable, such as spoken language or pedagogical language. Future research will aim to overcome some of these limitations.

3.2 CANDIDATE COLLECTION AND ANNOTATION

The previous study described in Section 2.2 observed that instances of mentioned language are relatively sparse in Wikipedia article text, occurring on average less often than once per article. Since hand annotation was a necessary step in creating the ML corpus, some heuristics were used to gather a rich set of mentioned language candidates.

Articles were randomly selected from English Wikipedia's most current article revisions, and heuristic filtering began at this level. Disambiguation pages were excluded from further examination, since they tend to be repetitive in structure and wording. Inside of articles, text from tables and common end sections (i.e., "Sources", "References", "See also", and "External links") also was excluded, since text from those sources was frequently observed to be non-sentential. The remaining article text was then segmented into sentences using NLTK's [11] Punkt sentence tokenizer. Those sentences that contained stylistic cues (bold text, italic text, or text between double quote marks) were retained, and all others were discarded. Applying this procedure to 3,831 articles produced a set of

22,071 sentences, which in turn contained 28,050 instances of text highlighted by stylistic cues[3].

Initial examinations of these remaining sentences suggested that mentioned language occurred in fewer than one in ten of them, and an additional heuristic was applied before manual annotation commenced. Using observations from the previous 171-sentence corpus, sets of "mention-significant" nouns and verbs were gathered. The appearance of a word from these sets near highlighted text signaled that the highlighted text was likely to be mentioned language. The procedure to gather these words was informal and manual, and a few potential mention-significant words (notably the verb *be*) were rejected because their great frequency reduced their significance as indicators. The eleven selected nouns and twelve selected verbs are listed below. The reader may note that most of the nouns refer to linguistic entities, while most of the verbs can serve as relational predicates or refer to speech acts:

Mention nouns: letter, meaning, name, phrase, pronunciation, sentence, sound, symbol, term, title, word

Mention verbs: ask, call, hear, mean, name, pronounce, refer, say, tell, title, translate, write

Words in the sentences were part-of-speech tagged and stemmed, again using tools from NLTK. The sentences were then filtered for those in which a mention word occurred (respecting the part of speech of its set) in the three-word phrase preceding text highlighted by a stylistic cue. This resulted in a set of 898 sentences, which in turn contained 1,164 instances of highlighted text. This set of instances was named the *ML-0* set.

Manual annotation of mentioned language then commenced. To eliminate possible biases, all three stylistic cues were substituted with pairs of asterisks (delimiting the beginning and end of highlighted text) prior to inspection. A human reader who was well-acquainted with the detection of mentioned language considered each instance in the ML-0 set and decided if it qualified by reading the sentence that contained it and applying the rubric from Section 2.1. 1,082 instances were deemed to be mentioned language, and this set was named the *ML-1* set, which also serves as the ML Corpus. This figure suggests that the heuristics leading to the creation of the ML-0 set have approximately 93%

[3] henceforth referred to as "highlighted text", for simplicity

precision for retrieving mentioned language, though their recall has not yet been measured.

3.3 Reliability and Consistency

Another limitation of the ML corpus is the lack of participation from multiple readers. To explore the possible impact of this, two additional human readers worked separately (from each other and from the primary reader) to annotate a 30-instance subset of the ML-0 set. These readers were also well-acquainted with the detection of mentioned language. Half of the 30 instances were selected from those annotated by the primary reader as mentioned language, and half were selected from those annotated as not. With that condition, the instances were randomly chosen from the ML-0 set, shuffled, and then distributed to the additional readers.

All three readers produced the same annotation for 25 of the instances, and on each of the remaining five, the additional readers differed with each other. (Since the annotation scheme was binary, this meant that one additional reader agreed with the primary reader and one disagreed). The kappa statistic was 0.779. These results were taken as a mild indication of reliability and consistency of the annotations in the ML corpus, while a second effort is currently underway to provide multiple annotations for all instances.

4 Results and Discussion

This section will present some notable findings distilled from the ML-0 and ML-1 sets. Particular attention was given to the precision of the heuristics used to create the ML-0 set. The combination of heuristics performed better (at 93% precision overall) than had been expected, with some standout performances from specific mention words and stylistic cues.

Below, Table 1 shows the frequency of mention words in the three-word phrases preceding each instance (an instance being a string of highlighted text) in the ML-0 set. Mention words were only counted if they appeared as their set-appropriate parts of speech. In the tables in this section, the precision shown is the percentage of those instances

deemed by the primary human reader to be mentioned language and thus placed in the ML-1 set.

Table 1. Frequencies of mention nouns (n) and verbs (v) in the three words preceding each instance in the ML-0 set, with their precisions for retrieving mentioned language.

Mention word	Frequency	Precision (%)
call (v)	349	98.6
name (n)	153	98
name (v)	89	94.4
say (v)	86	94.2
term (n)	79	98.7
title (n)	72	84.7
title (v)	64	96.9
word (n)	55	100
write (v)	52	50
mean (v)	39	100
refer (v)	35	85.7
meaning (n)	20	100
translate (v)	20	20
phrase (n)	18	100
symbol (n)	10	80
pronounce (v)	8	100
tell (v)	7	71.4
letter (n)	6	33.3
pronunciation (n)	4	100
ask (v)	4	75
sentence (n)	3	33.3
hear (v)	3	0
sound (n)	1	0

As shown, the verb *call* and the noun *name* stood out as the most common of the mention words, with all others forming a relatively smooth tail of descending frequency. These top two are intuitive, the informative and descriptive purposes of Wikipedia articles. Both words also had substantially above-average precision. *Word* (n), *meaning* (n), *phrase* (n), *pronounce* (n), and *pronunciation* (v) all had perfect precision, though they appeared less frequently. However, following the multiple-reader experiment in Section 3.3, it was discovered that *meaning* instances were particularly difficult to classify, generating some debate among the participants. Finally, an observant reader may note that the frequencies in Table 1 sum to 1,177 instead of 1,164 (the

size of the ML-0 set). This is because 13 instances had more than one mention word in the preceding three-word phrase. All 13 of these instances were annotated as mentioned language.

Although stylistic cues were hidden from the readers while they annotated instances, data on the cues was retained. Table 2 below breaks down their frequencies and precisions.

Table 2. Frequencies of stylistic cues in the ML-0 set and their precisions for retrieving mentioned language.

Stylistic cue	Frequency	Precision (%)
double quote	601	96.7
italic	427	86.4
bold	136	97.1

Double quote marks had the highest frequency, and the reason was first assumed to be frequent quotation (in the sense of speech reporting, for example) in Wikipedia. However, as Table 5 will show, that was probably not the case. Italics had by far the lowest precision. 23 of the 58 non-mention italic instances had *write* (v) as a preceding mention word, which conjures a common construction (as in "Dickens wrote *Great Expectations*…") that does not involve mentioned language. Bold had both the highest precision and lowest frequency. It is worth noting that Wikipedia articles, by convention, contain the article subject in bold text in the first sentence.

Prior to analysis, it was hypothesized that the proximity of a mention word to highlighted text increases its likelihood of being mentioned language. Table 3 shows this hypothesis to be true, albeit in the limited three-word window that was examined. Also shown are overall frequencies and precision percentages (weighted by frequencies) for nouns and verbs.

Table 3. Frequencies of mention nouns and verbs in the three words preceding highlighted text (e.g., word position 1 is the word just before the highlighted text), with their precisions for retrieving mentioned language.

Noun/Verb	Frequency		Precision (%)	
position	Noun	Verb	Noun	Verb
1	281	458	98.6	97.2
2	89	179	91.0	85.5
3	51	119	76.5	84.0
overall	421	756	94.3	92.4

There appears to be a strong correlation between proximity and precision, though *proximity* in this data does not account for the grammatical structure of corpus sentences, which will deserve examination in future research. A mention verb directly preceding highlighted text was by far the most common combination. Overall, mention nouns had a slightly greater precision than mention verbs.

Finally, Table 4 shows the most common mention word-stylistic cue combinations in the ML-1 set.

Table 4. The ten most frequent word and stylistic cue combinations in the ML-1 set, with their percentages of the total (1082) instances. Out of 69 possible different word-cue combinations, 59 were observed.

Word	Cue	Frequency	% of total
call (v)	d. quote	151	14.0
call (v)	italic	133	12.1
say (v)	d. quote	74	6.8
name (n)	italic	60	5.5
name (n)	d. quote	56	5.2
call (v)	bold	53	4.9
term (n)	d. quote	45	4.2
name (v)	d. quote	39	3.6
title (v)	italic	36	3.3
title (n)	italic	32	3.0

The prevalence of *call* (v) is once again apparent, as it appears in the two most common combinations. Double quote marks with *say* is the third most common combination, which matches earlier intuitions on quotation, but the same stylistic cue appears frequently with *call* (v), *name* (n), *term* (n), and *name* (v) as well. Bold makes only one appearance in the top ten, in combination with the previously mentioned *call* (v). These ten combinations account for only 17% of the combinations observed but 62.6% of all instances in the ML set.

Overall, it is believed that these results validate the heuristics that were used to collect candidate instances. They also seem to confirm that Wikipedia is a fertile source of mentioned language, as the instances exhibit a variety of different constructions. Given the size of Wikipedia and the current methods for collecting candidates, future expansion of the ML corpus will be possible and productive.

5 Related Work

Wikipedia's emerging utility as a corpus is well-documented in the literature. A few related uses of Wikipedia include named entity recognition [12], syntactic parsing [13], and lexical semantics [14] among many others. Ytrestøl *et al*. [15] previously noted the relationship between stylistic cues in Wikipedia and the use-mention distinction, though this observation was incidental to their focus on the automatic extraction of sub-domains of articles. The use of stylistic cues described in this paper appears to be unique.

As mentioned in the introduction, little previous study exists of the use-mention distinction as it appears in linguistic corpora. Notably, Anderson et al. [16] gathered by hand a corpus of metalanguage in human dialogue using a subset of the British National Corpus. Their annotation scheme applied to sentence-level utterances, and focused on metalanguage as used to maintain grounding and recover from perturbations (e.g., misunderstandings and interruptions). Mentioned language generally—perhaps always—requires metalanguage to frame it, and it is likely that many instances of the phenomenon were gathered for such a corpus. However, the annotation scheme was not designed to address mentioned language either as a distinct category (it fit partially into several) or as a phenomenon in the structure of a sentence. This difference in focus, as well as the difference in language context, made the findings of the Anderson corpus and the present one substantially different.

6 Future Work

The ML corpus is significant as the first of its kind, but it has some limitations that will require further work. The heuristics used to identify candidates have high precision, but their recall has not yet been measured. However, it is anticipated that the ML corpus is large and varied enough to provide a basis for "bootstrapping" the detection of mentioned language outside of the heuristics presently used. This will be done through a more detailed examination of syntactic and lexical patterns in the sentences contained in the ML corpus. WordNet [17] is expected to be a useful resource for expanding the sets of mention nouns and verbs currently used. To expand the data on inter-annotator agreement, Amazon's Mechanical Turk service is being considered as a

source for additional participants. Although such participants would not be experts, the service was previously tested by Snow et al. [18] with similar annotation tasks and was found to be fairly accurate. Preliminary tests of Mechanical Turk for a use-mention annotation task have yielded positive results.

An eventual goal of this research is the automatic detection of mentioned language in a variety of contexts outside of Wikipedia. Although some retraining of a Wikipedia-based classifier is likely to be necessary, it is hypothesized that a core set of metalinguistic cues are shared across different language contexts. Additionally, previous research has shown that statistically-trained English parsers tend to make egregious errors when faced with even simple forms of mentioned language [9], and rectifying such errors is a further goal. Both goals are motivated by the considerable importance of the phenomenon in the human use of language.

7 Conclusion

This study has created a corpus of instances of mentioned language, using the particularly suitable properties of Wikipedia article text. The ML corpus is a unique resource that should provide a springboard for future research on the use-mention distinction and its relevance to a variety of language processing tasks. Results discussed in this paper show that patterns exist in mentioned language which can be utilized to expand the corpus and to apply machine learning techniques to it. This will eventually benefit both our understanding of the use-mention distinction and our ability to build language systems that recognize and exploit it.

Acknowledgments. Several conversations with Donald Perlis and Scott Fults were valuable in formulating this study. This research was supported in part by the US Office of Naval Research and the Air Force Office of Scientific Research.

References

1. García-Carpintero, M.: The deferred ostension theory of quotation. Noûs. 38, 674–692 (2004).

2. Cappelen, H., Lepore, E.: Varieties of quotation. Mind. 106, 429 -450 (1997).
3. Davidson, D.: Quotation. Theory and Decision. 11, 27-40 (1979).
4. Saka, P.: Quotational Constructions. Belgian Journal of Linguistics. 17, (2003).
5. Anderson, M.L., Okamoto, Y.A., Josyula, D., Perlis, D.: The Use-Mention Distinction and its Importance to HCI. In Proceedings of the Sixth Workshop on the Semantics and Pragmatics of Dialog. 21--28 (2002).
6. Perlis, D., Purang, K., Andersen, C.: Conversational adequacy: mistakes are the essence. International Journal of Human-Computer Studies. 48, 553-575 (1998).
7. Saka, P.: Quotation and the use-mention distinction. Mind. 107, 113 -135 (1998).
8. Miller, M.J.: A view of one's past and other aspects of reasoned change in belief, http://portal.acm.org/citation.cfm?id=164828, (1993).
9. Wilson, S.: Distinguishing Use and Mention in Natural Language. Proceedings of the NAACL HLT 2010 Student Research Workshop. pp. 29–33Association for Computational Linguistics, Los Angeles, CA (2010).
10. Maier, E.: Mixed quotation: between use and mention. Presented at the Logic and Engineering of Natural Language Semantics Workshop, Miyazaki (2007).
11. Garrette, D., Klein, E.: An extensible toolkit for computational semantics. Proceedings of the Eighth International Conference on Computational Semantics. pp. 116-127Association for Computational Linguistics, Tilburg (2009).
12. Balasuriya, D., Ringland, N., Nothman, J., Murphy, T., Curran, J.R.: Named entity recognition in Wikipedia. Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources. pp. 10-18Association for Computational Linguistics, Suntec, Singapore (2009).
13. Honnibal, M., Nothman, J., Curran, J.R.: Evaluating a statistical CCG parser on Wikipedia. Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources. pp. 38-41Association for Computational Linguistics, Suntec, Singapore (2009).
14. Zesch, T., Müller, C., Gurevych, I.: Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. Proceedings of LREC '08. European Language Resources Association (ELRA), Marrakech, Morocco (2008).
15. Ytrestøl, G., Flickinger, D., Oepen, S.: Extracting and annotating Wikipedia sub-domains. Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories. (2009).
16. Anderson, M.L., Fister, A., Lee, B., Wang, D.: On the frequency and types of meta-language in conversation: a preliminary report. 14th Annual

Conference of the Society for Text and Discourse. (2004).
17. Fellbaum, C.: WordNet: an electronic lexical database. MIT Press, Cambridge (1998).
18. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast--but is it good?: evaluating non-expert annotations for natural language tasks. Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 254-263. Association for Computational Linguistics, Honolulu, Hawaii (2008).

SHOMIR WILSON
DEPARTMENT OF COMPUTER SCIENCE,
UNIVERSITY OF MARYLAND AT COLLEGE PARK,
COLLEGE PARK, MARYLAND 20742, USA
E-MAIL: <SHOMIR@UMD.EDU>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 155–173
RECEIVED 01/11/10 ACCEPTED 02/12/10 FINAL 12/02/11

Evaluation of a Unified Dialogue Model for Human-Computer Interaction

HUI SHI,[1,2] CUI JIAN,[1] AND CARSTEN RACHUY[1]

[1] *Universität Bremen, Germany*
[2] *DFKI Bremen, Germany*

ABSTRACT

This paper reports our work on evaluating the task success of a dialogue model developed by a unified dialogue modeling approach for human-computer interaction, which combines an information state based dialogue theory and a state-transition based modeling approach at the illocutionary level. As an application, the unified dialogue model has been integrated into a multimodal interactive guidance system for hospital visitors. An experiment with 12 subjects has been carried out. Using the collected dialogue data we have evaluated the task success of the dialogue model by the Kappa coefficient. The results show that the unified dialogue model is highly effective and provide several valuable improvements for the further development as well.

KEYWORDS: *human-computer dialogue, dialogue act, illocutionary structure, information state, dialogue system evaluation, formal methods*

1 INTRODUCTION

Generalized Dialogue Modeling (cf. [14, 8, 12]) and *Information State* based dialogue theories (cf. [15, 5, 2, 4, 7, 16]) are the two most important approaches to develop dialogue models. Generalized dialogue models are based on recursive transition networks. These models consist of pattern-based accounts of dialogue structure at the illocutionary level and

therefore, are independent of utterance content or other direct surface indicators. Information state theories, on the other hand, offer a powerful basis for interaction analysis and practical dialogue system construction. However, such information state based dialogue models are difficult to manage, to extend and to reuse. Although it has been suggested that applying generalized dialogue models to information state based accounts could eliminate some of the perceived problems, there have only been preliminary researches to date [18, 8]. In Lewin [8], for example, recursive transition networks were applied to model Conversational Game Theory by combining dialogue grammars with discourse planning.

The unified dialogue modeling approach introduced in this paper combines the information state based dialogue theory discussed in [16, 7] and the generalized dialogue modeling approach proposed in [14, 12]. Specifically, unified dialogue models extend generalized dialogue models by introducing *context-sensitive* transitions, which allow for direct integration with information state management. A unified dialogue model is represented as the traversal of a state-transition network with arcs denoting context-sensitive transitions and nodes denoting dialogue states. In addition to the allowed dialogue action, each context-sensitive transition is associated with a set of *conditions* under which the dialogue action can be taken and a set of *update rules* for updating the information state after performing the dialogue action.

As emphasized in [11, 12], the separation of illocutionary structures from the information state-based modeling enables the formal analysis and comparison of illocutionary structures by applying well-established techniques from the formal methods community of computer science. In this paper, we focus on the evaluation of unified dialogue models. The *Kappa coefficient* [13, 3] has been proposed as a standard measure of reliability and task success ([17]) for evaluating spoken dialogue systems. Therefore, we apply it to evaluate how well human users can be supported by the unified dialogue model implemented in a multimodal dialogue system for guiding visitors in hospital environments. For this purpose we carried out an experiment with 12 people and collected 272 dialogues.

This paper is organized as follows: Section 2 introduces the unified dialogue modeling approach, which has been applied to develop a unified dialogue model for a practical multimodal dialogue system presented in Section 3. Section 4 describes the experiment and the collected dialogue data, which are then used to evaluate the unified dialogue model by the Kappa coefficient in Sections 5 with respect to the measure of task success. The evaluation results and corresponding improvements are

discussed in Section 6. Finally, Section 7 concludes with the outline of future work.

2 A UNIFIED APPROACH FOR DIALOGUE MODELING

The unified modeling approach takes as a starting point existing researches on the generalized dialogue modeling at the illocutionary level using Recursive Transition Networks (RTNs) [14]. Unlike finite state models, the RTNs employed here capture more abstract dialogue models which depict discourse patterns in illocutionary force terms only – without reference to propositional content or other direct surface indicators. Fig. 1(a) depicts a transition diagram named *Assert(A,B)* initiated by a dialogue participant, say A, and responded to by B. The darkened circles denote final states. This generalized transition diagram is initiated by A's dialogue move of type *assert*. The possible responses from B are threefold: B agrees with the assertion (*B.agree*), accepts it (*B.accept*) or rejects it (*B.reject*). To note that, the transition diagrams *Ask(B,A)* and *Assert(B,A)* are used to enable B to ask some question(s) before reacting to A's request, or to give possible reason(s) by a rejection, and are not presented here in detail.

Generalized dialogue models such as the one depicted in Figure 1(a) are non-deterministic models, where more than one dialogue move is able to trigger state transitions starting from one state. The decision as to which transition should be activated naturally depends to a certain extend on B's pragmatic domain knowledge. To take domain knowledge into account, thus to solve such nondeterministic transitions, *conditional transitions* are introduced in unified dialogue models. A conditional transition can be activated only if its conditions are satisfied. Let *checkAssert* be an operation provided by B's domain component, which takes an assertion as a parameter and returns *true* if B's knowledge matches the assertion; or *false* if the assertion conflicts with some of B's knowledge (in that case, the transition diagram *Assert(B,A)* will be activated to explain the reason for B's rejection); or *added*, if the assertion can be added by B as a new element to the knowledge base. A deterministic transition diagram for the example is now shown in Figure 1(b), where a is assumed to be the assertion made by A.

Although conditional transition models as shown in Fig. 1(b) capture the illocutionary structure of dialogues and are deterministic as well, they do not provide mechanisms to integrate dialogue context and history. Therefore, they do not reflect dialogue participants' attitudinal state

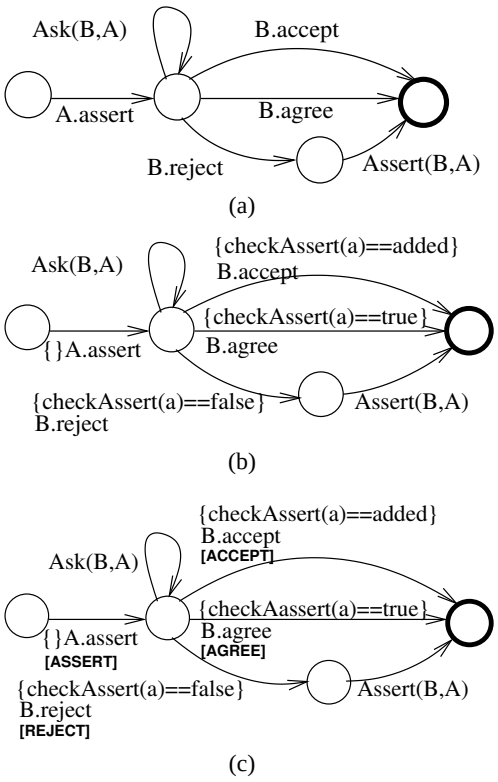


Fig. 1. Three transition diagrams: (a) non-deterministic assertion, (b) deterministic assertion, and (c) deterministic assertion with update rules

along with the behavioral mechanisms for dialogue progression and the dynamic update of attitudinal states over time. As indicated earlier, information state based approaches of dialogue models [9, 15, 4] and dialogue management [16, 7] focus on the modeling of dialogue contexts and participants' attitudinal states, apart from that they do not capture the structural features of dialogues. Thus, merging these two approaches is valuable, so that the basic formalism of the conditional transition models is extended by introducing a mechanism to interface with information state.

Since generalized dialogue models already capture structural features of dialogue moves, some of the typical *structural elements* in the infor-

mation state based accounts, e.g., *AGENDA* for keeping the planed dialogue acts in Ginzburg and Larsson's models, become unnecessary, hence the information model can be simplified considerably. In unified dialogue models, each transition can be associated with one or more update rules for updating the current information state if needed before proceeding to the next state. As usual, an update rule consists of a name, a set of preconditions and a set of operations on information states. To illustrate this model extension, we again take the transition diagram *Assert(A,B)* as an example and show it in Fig. 1(c). After dialogue participant A makes an assertion, the update rule *ASSERT* will be applied to update the information state, such that the new assertion can be integrated into the current information state. Similarly, B's transitions of *accept*, *agree* and *reject* can change the information state by the corresponding update rules.

Finally, a *unified dialogue model* is a pair $\langle \mathcal{G}, G_0 \rangle$ of a transition network $\mathcal{G}$ with a set of extended recursive transition diagrams and a main diagram $G_0 \in \mathcal{G}$. Each transition may contain some conditions and information state update rule(s). Specifically, if a dialogue is in the start state of a transition whose conditions are satisfied, the corresponding dialogue move is then enabled and the information state is updated by its update rules, and the dialogue will move to its goal state.

3 MIGHE: A MULTIMODEL INTERACTIVE GUIDANCE FOR HOSPITAL ENVIRONMENT

MIGHE is a multimodal interaction system developed for guiding people in public areas such as hospitals. Fig. 2 shows the overall MIGHE architecture. This section focuses on the development of a unified dialogue model and its integration into the dialogue system. The unified dialogue model is implemented within the two components: the *dialogue controller* and the *information state manager*. The *clinic database manager* provides the dialogue controller with necessary information about application environment. The dialogue controller manages the communication between various system components, and controls the dialogue process according to the dialogue model together with the information state manager. The guidance system supports both natural language inputs and touch events, but in the experiment presented in Section 4 only the natural language input channel is enabled.

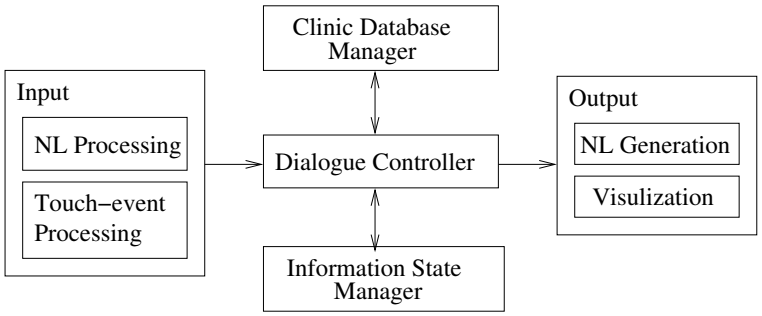


Fig. 2. The overall architecture of the dialogue system

3.1 *The Unified Dialogue Model*

The dialogue model implemented in MIGHE is developed according to the unified dialogue modeling approach introduced in Section 2. In this paper we focus on the task orientated dialogues and disregard communication problems like failures by speech recognition or misunderstanding. Generally, these problems can be treated by extending the dialogue model. The information state structure consists of two parts: *LM* for keeping the latest dialogue move and *CONTEXT* containing a list of contexts of active (sub-)dialogues. In this application, the possible contexts are of the types: *department*, *person*, or *room*, which provide context information for integrating user's dialogue moves, for example, "go to *a room* of *a known department*", or "request for information of *a person* in *a department*".

The unified dialogue model consists of four extended transition diagrams with the main diagram *Dialogue(S,U)*, see Fig. 3. After a system's initializing *request* (Fig. 3(a)), the user can instruct the system to find some visiting goals by utterances with the dialogue act *instruct*, or ask the system to find certain information by *request* (see *Dialogue(U,S)* in Fig. 3(b)). The network *Response(S,U)* (Fig. 3(c)) specifies all deterministic system responses after getting an input from the user according to its domain knowledge and the current information state. If the requested information or instructed goal does not exist, the user's input is *rejected*, probably with a reason if the relevant information is available. If it is found unambiguously, the user is *informed* and asked whether he/she would like to take the found place as a destination in case the last user input is an instruction. However, if more than one possibility are found, a subdialogue

is started by the system for asking the user to make a *choice*. Finally, *Response(U,S)* (Fig. 3(d)) describes possible *nondeterministic* user reactions to a system's request. Moreover, each dialogue move issued by a user in the dialogue model is associated with the name of an update rule.

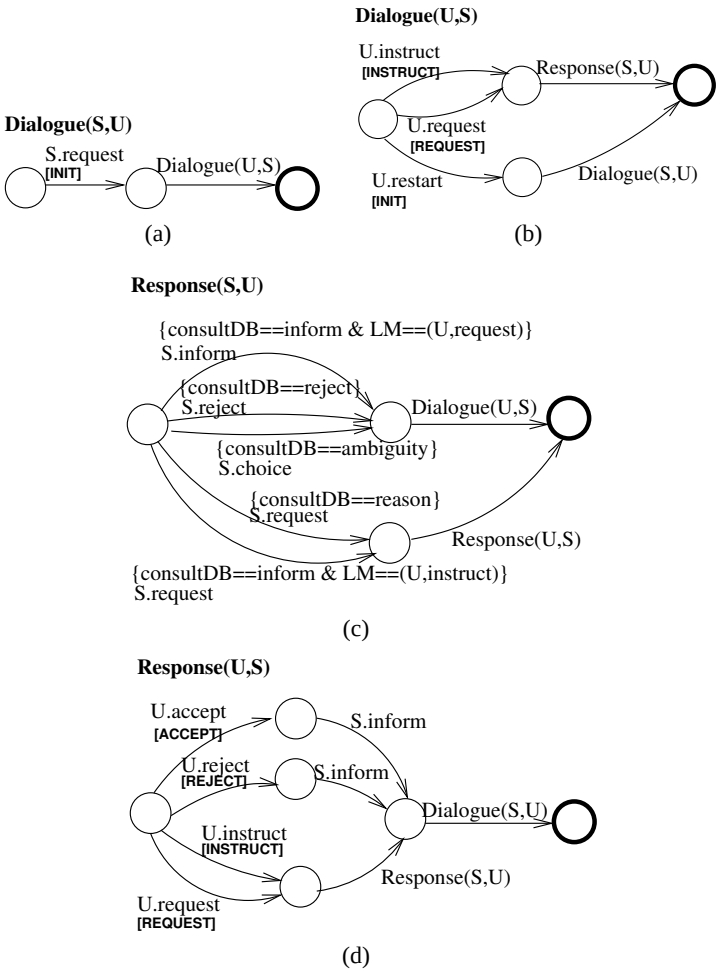


Fig. 3. The unified dialogue model: (a) the main transition diagram, (b) the transitions issued by the user, (c) the system's responses and (d) the user's response

3.2 *Integrating the Unified Dialogue Model into the Dialogue System*

The implementation of the unified dialogue model is carried out in two major steps. In the first step, a set of update rules as required by the dialogue model is implemented for the component *information state manager*. Five update rules are needed in the unified dialogue model (see Fig. 3). The following shows the rule INSTRUCT as an example. Suppose that *context* and *dest* are two operations to identify the context and destination contained in an input, respectively. The context and destination of "I'd like to go to Mrs. Angelika Fromm in Gastroenterology", for example, are "Gastroenterology" and "Mrs. Angelika Fromm". If the current instruction contains context information, i.e., the user gives the context in his/her instruction explicitly, then the new context will be added to *CONTEXT*, otherwise, the most actual context in *CONTEXT* (or *top*(*CONTEXT*)) is used to complete the current instruction. The other rules are defined accordingly.

RULE: INSTRUCT
PRE: if *context*(m)!=*null* then c = *context*(m)
else c = *top*(*CONTEXT*), d = *dest*(m)
EFF *LM* = (U, *instruct*),
if *context*(m)!=*null* then *CONTEXT* = *add*(*CONTEXT*, c)

The second step is the development of the control mechanism of the component *dialogue controller*, which is based on the dialogue state transitions at the illocutionary level specified by the dialogue model. As the unified dialogue model defines a clear illocutionary structure represented by a set of extended recursive transition diagrams, it can be specified with mathematically well-founded methods straightforwardly, e.g., the well-established technique from the formal methods community of computer science *Communicating Sequential Processes* (CSP). The CSP language provides mechanisms for specifying the communication and synchronization of two or more processes consisting of sequential actions. The essential value of CSP is the ability to subject formal specifications that are well founded in mathematical logic to enable powerful analysis using mechanized theorem provers and model checkers (cf. [12]). Although the CSP language, its mathematical foundations and its many possible applications within the Formal Methods Community have been widely investigated [6, 10], applying these techniques to dialogue modeling, specification and analysis builds up a novel area of application. In the following we will briefly introduce the specification of the unified dialogue model presented in Section 3.1 using CSP.

The first CSP process *DialogueUS* in Fig. 4 specifies the transition network *Dialogue(U,S)*, where $\rightarrow$ and [ ] are two CSP operators necessary for the present specification. $\rightarrow$ defines the sequential occurrence of dialogue moves in a process, and [ ] arbitrary selection between several possibilities. The CSP events representing abstract dialogue moves have the form $p.a$, where p is the name of a communication channel and a the dialogue act associated with it. For example, *user.instruct* means getting an input with the dialogue act *instruct* from the user, *is_out.instruct* sending the dialogue act *instruct* to the information state manager, such that the information state can be updated using the context contained in the current input. Obviously, the specification reflects the model structure very well. The second CSP process *ResponseSU* invoked by the first one in Fig. 4 specifies the transition network *Response(S,U)*, in which the *latest dialogue move* kept in the information state is needed. In the specification *ResponseSU* the conditions related to *consultDB* are specified by four database input *db_in* events: *reject*, *reason*, *inform* and *ambiguity*. Also the CSP specification of *Response(U,S)* reflects the network structure straightforwardly.

```
DialogueUS =
    user.restart -> is_out.init -> DialogueSU
 [] user.instruct -> is_out.instruct -> ResponseSU
 [] user.request -> is_out.request -> ResponseSU

ResponseSU = db_out -> (
    db_in.reject -> system.reject -> DialogueUS
 [] db_in.reason -> system.request -> ResponseUS
 [] db_in.inform -> is_in?lm ->
   ( (lm==request) & (system.inform -> DialogueUS)
   [](lm==instruct) & (system.request -> ResponseUS))
 [] db_in.ambiguity -> system.choice -> DialogueUS)
```

Fig. 4. Two CSP specifications

Based on the CSP specifications the model-checker FDR [1] is applied to generate the state machine. After implementing the communication channels between the dialogue controller and the other system components, the state machine can control the state transitions according to communication events.

4 THE EXPERIMENT

In order to explore how well the dialogue interaction between human and the dialogue system is assisted by the unified dialogue model, an evaluation with 12 participants was carried out. Each subject had to undergo two test phases: learning and testing:

- In the learning phase each participant was given a brief introduction to the test procedure, so that they could get to know the way how to dialogue with the system, and what kinds of verbal and textual feedbacks the system provides. Furthermore, they were asked to accomplish several sample tasks.
- In the test phase each participant had to go through three subphases, each of which contains several tasks belonging to a predefined category. In the first subphase, several pieces of information describing a destination (e.g. a person's name, a department or a room number) were given and the participant should tell the system to go there. In the second subphase, pieces of information were given as well, but this time the participant was asked to find out certain information, e.g. where a certain person works or what department a room is in. In the third subphase scenarios like "you are hungry and would like to eat something" were described, and the participant was asked to negotiate with the system on an appropriate destination.

The dialogue system used in the experiment was a networked software application that connected two computers: the *guidance assistant* on one computer and the *input system* on the other. The *input system* was controlled by a human operator who entered the user utterances and acted as a speech recognizer. The *guidance assistant* contains the components *clinic database manager*, *output*, *information state manager* and *dialogue controller*, and the unified dialogue model is the key of the *information state manager* and *dialogue controller*. As a result, the whole test run was simulated as if the participant communicates with the system in natural language directly, but removing possible distractions that might have been introduced by speech recognition, in order to focus on the evaluation of the unified dialogue model. Although a human operator acted as the speech recognizer, our experiment was not a usual "Wizard of Oz" experiment, since the *guidance assistant* ran automatically.

Since the experiment was run with native German-speaking participants, we present in the following the English translations of several example dialogues collected in the experiment. Most of the dialogues turned

out to be unproblematic, as the following example shows. The task of the example contained the destination “Rasmussen” and “Room number 1322”. The room number in the user’s *instruction* (*D_01_01*) was incorrect, thus the system *rejected* it in *D_01_02* with a reason. In *D_01_03* the user provided the room number again, which is interpreted as a new *instruction* integrated with the context *person* provided in the previous instruction and kept in *CONTEXT* of the information state. This time the system found the person and his office, and then responded with a *request* such that the user can confirm or reject it.

D_01_01 U I’d like to go to Mrs. or Mr. Rasmussen in room number 1320.
D_01_02 S I did find Rasmussen, but not in room 1320.
D_01_03 U I’d like to go to room 1322.
D_01_04 S Would you like to go to Mr. Per Rasmussen in room 1322, the office of chief physician?
D_01_05 U Yes.

However, there are dialogues turned out to be problematic. The utterances from *D_02_01* to *D_02_04* fit well to the model, but the utterance *D_02_05* is an user *request*, for which no proper system response is defined. Therefore, the system alternatively offered the *information* about all persons (*D_02_06*). Again the user *requested* the system to name a person with the given name in *D_02_07*, the system found a person with the name, and *requested* the user to confirm, since it is the only person with the last name “Brückner”. The system utterance *U_04_08* did not match the user’s request in *D_02_07*, showing an unnatural dialogue segment.

D_02_01 U I’d like to go to Brückner.
D_02_02 S Would you like to drive to the doctor Mr. Wolfgang Brückner?
D_02_03 U I need more information about him.
D_02_04 S Mr. Wolfgang Brückner is a doctor in the accident surgery. You find him in room 1711.
D_02_05 U Is there an other person with the name Brückner?
D_02_06 S I found all together 32 persons and they are sorted according to their last names.
D_02_07 U Name a person with the name Brückner.
D_02_08 S Would you like to drive to the doctor Mr. Wolfgang Brückner?
D_02_09 U If there is no other person with the name Brückner, then yes.

5 DIALOGUE MODEL EVALUATION CONCERNING TASK SUCCESS

One aim of our experiment was to investigate the task success of the unified dialogue model in a practical dialogue system. Specifically, in this section we are going to evaluate how the system's deterministic behavior (see Fig. 3(c)) influences the task success. Success at the task of a dialogue in our context is measured by how well the dialogue model supports users to complete dialogue tasks and therefore, we apply the Kappa coefficient [13, 3, 17] approach, similar applications can be found in the literature, such as the evaluation of two train timetable information agents in [17].

First, we define a set of *attribute values* for each task. As shown in Fig. 3(d) the unified dialogue model allows a user to make a dialogue move with an *instruction* like "take me to $\cdots$", a *request* like "tell me about $\cdots$", an *accept* like "yes" or a *reject* like "no" after a system's utterance. Each user's dialogue move may contain some content information, also called *attribute values*, of a person's name, a room number and so on. Tab. 1 summarized the set of all relevant attributes.

Table 1. The set of attributes

attribute name	identifier	description	example
first name	FN	first name of a person	Wolfgang
last name	LN	last name of a person	Brückner
gender	G	gender of a person	M
profession	P	profession of a person	Doctor
room number	RNr	number of a room	1711
room type	RT	type of a room	station room
meta room type	MRT	predefined meta type of a room	eating-related
station	F	name of a hospital station	accident surgery

Since different tasks contain different data and have different goals, each task has a set of expected dialogue acts and attribute values, such as the *attribute value matrix* (AVM) in Tab. 2 for the task, in which the participants were asked to go to a person with the last name "Brückner" (see the example dialogue *D_02* in Section 4). Each expected dialogue act-attribute pair is associated with an actual value, which reflects the fact that a unified dialogue model contains a state transition based structure at the illocutionary level and an information state management processes.

With the attribute value matrix we can develop the confusion matrix for the collected dialogue data of that task (see Tab. 3).

Table 2. An example of value matrices for dialogue acts and attribute values

dialogue act	attribute	actual values
instruct	LN	Brückner
	G	M, F
accept	LN	Brückner
	FN	Wolfgang
	P	Doctor
	G	M

Table 3. An example confusion matrix

data		instruct				accept								other	sum
		LN		G		LN		FN		P		G			
		E	NE	E	NE	E	NE	E	NE	E	NE	E	NE		
instruct	LN	12												4	16
	G			9											9
accept	LN					12									12
	FN							11							11
	P									9					9
	G											11			11

The values in the confusion matrix are obtained by comparing the dialogue moves issued by the participants and the expected attribute values of each task specified by a AVM. A user dialogue move may contain expected or unexpected information with respect to the attribute values defined in the AVM for a dialogue task, so we use “E” and “NE” in confusion matrices to denote such situations. Values in the “other” column record the number of undefined dialogue moves occurred in the dialogue data. Hence, these confusion matrices capture not only expected dialogue situations, but also unexpected and undefined situations.

Given a confusion matrix, the success at reaching dialogue goals is measured with the Kappa coefficient [13, 3, 17]: $\kappa = \frac{P(A)-P(E)}{1-P(E)}$, where $P(A)$ is the proportion of times that the dialogue moves agree with the

attribute values and $P(E)$ is the proportion of times that the dialogue moves are expected to be agreed by chance. In our case,

$$P(A) = \frac{\sum_{i=1}^{n} M(i,E)}{T}, \qquad P(E) = \sum_{i=1}^{n} \left(\frac{M(i)}{T} \right)^2$$

where $M(i,E)$ is the value in an expected column of row i, T is the sum of all user dialogue moves, and $M(i)$ the sum of the user dialogue moves in row i.

Since our goal is to find out how well the dialogue model implemented in the dialogue system supports various types of tasks, instead of individual tasks, we first calculate the Kappa coefficient for each type by the confusion matrix combining all the confusion matrices of the tasks in that type. The first type contains 13 tasks with 149 dialogues, the second type 3 tasks with 35 dialogues, the third type 8 tasks with 88 dialogues. Since the third type contains the second type implicitly, only three tasks were taken in the experiment for the second type. Finally, the three confusion matrices of the three individual task types are combined to a single confusion matrix for computing the total Kappa coefficient. The results are presented in Tab. 4.

Table 4. The task type dependent and independent Kappa coefficients

task type	type I	type II	type III	type I, II, III
Kappa coefficient	$\kappa_1 = 0.99$	$\kappa_2 = 0.85$	$\kappa_3 = 0.82$	$\kappa = 0.94$

6 Discussion of Evaluation Results and Improvements

From the Kappa coefficients calculated in table 4, we can see that the unified dialogue model well supports the three types of dialogue tasks: the Kappa coefficients κ_1, κ_2 and κ_3 are greater than 0.8 and the task type independent Kappa coefficient κ is 0.94.

Especially, the achievement of the dialogue model by the first type of tasks is obvious, since the Kappa coefficient almost equals 1. In all the 13 tasks of the first type the participants were asked to instruct the system to a destination described by several pieces of information about a person, a room or a department (see the dialogue example *D_01* in Section 4), and therefore, the dialogue goal of each task was given explicitly. On the

other hand, each dialogue goal in the third type was described implicitly, such as the goal of the task described by "you are hungry and would like to eat something" was "the patient dining-hall". From the fact $\kappa_1 > \kappa_3$ we can conclude that the unified dialogue model supports dialogues with a clearly defined goal much better than those without. Missing the negotiation of users' intention in the unified dialogue model is the key fact leading to this finding.

The analysis of the confusion matrices issues two drawbacks of the unified dialogue model: losing track of the current user dialogue move type and missing some unexpected dialogue move types. For example, the dialogue *D_03* itself was straightforward. The problem was that the system could not continue the dialogue after the utterance *D_03_03*, since it lost the information to decide whether the utterance is a *request* or an *instruct* move and therefore, the dialogue model cannot determine the next transition to be taken (see Fig. 3(c)). The problem can be solved by adding the *move context* into the information state, which keeps the current user dialogue act, in our case *instruct* or *request*.

D_03_01 U Which profession does Mr. Schulte practice?
D_03_02 S I have found several persons who match Mr. Schulte.
Which person do you mean?
D_03_03 U I mean Mr. Tobias Schulte.

Extending the illocutionary structure of the unified dialogue model is another step to improve the dialogue model. *D_04*, for example, contains the user dialogue move *D_04_02*, where the user told the system that he did not know the room number. This type of dialogue moves is not supported by the illocutionary structure of the dialogue model. Adding a new transition *U.inform* to the start state in the transition dialogue of *Response(U,S)* (see Fig. 3(d)) enables the dialogue model to handle such dialogue moves.

D_04_01 S Would you like to drive to room 1262, ECG 2, in the cardiology?
D_04_02 UI don't know the room number.

The refinements of the dialogue model by adding new information state elements and additional transitions have been applied to update the dialogue system. We believe that they will improve the task success of the unified dialogue model throughout. This has to be proved by a follow-up experiment.

Based on the evaluation results, we conclude that the unified dialogue model well supports users to dialogue with the hospital guidance system,

however, they cannot be used to measure the effectiveness of the whole dialogue system, since all the test runs were, with the assistance of a human operator[3], simulated as if the participants were conversing with the system in natural language directly, but removing possible distractions that might have been introduced by speech recognition. Comparing the audio data with the manual input data did not deliver any essential deviation that would affect task successes of any undergone dialogues. Therefore, our focus on evaluation of the unified dialogue model is maintained.

Unified dialogue models are constructed at the illocutionary force level, which naturally enables dealing with diversity situations. However, choosing the appropriate set of communicative acts is one important factor affecting the coverage of a unified dialogue model. Care must be taken on the one hand to avoid over-simplification to the point where the structural model collapses down to a two-state initiate-response network with jumps. Although these over-simplified models capture most dialogue situations, they are not useful for dialogue control or formal analysis of dialogue structure. On the other hand, models, as the one discussed in this paper, well reflect natural dialogue structures at the illocutionary level and still possess the context sensitive information state management that relies on domain specific communication. Diversity problems might occur when people dialogue with a system based on a too excessively designed unified dialogue model, but through appropriate design and careful evaluation possible diversities can be detected and the model can then be improved accordingly.

7 CONCLUSION

In this paper we applied the Kappa coefficient (κ) to evaluate the effectiveness of a unified dialogue model by task success, which combines a generalized dialogic structure at the illocutionary level and an information state based content manager. Specifically, three Kappa coefficients were calculated from the confusion matrices for three types of dialogue tasks using the 272 dialogues collected in an experiment with 12 participants. The results showed that the unified dialogue model well supports those dialogue tasks in general ($\kappa = 0.94$). Especially, tasks with an explicit defined dialogue goal (cf. $\kappa_1 = 0.99$). The experiment results also delivered useful findings for the improvement of the dialogue model. This

[3] We used only one operator in the whole experiment

paper has three major contributions. First, it showed the development of unified dialogue models in general and by an example. Second, we demonstrated how to evaluate unified dialogue models by combining dialogue acts with attribute values. Third, we applied the standard method, the Kappa coefficient, to evaluate a unified dialogue model.

To evaluate the improvement of the unified dialogue model according to the analysis of the experiment results, we are now carrying out a follow-up experiment. The collected dialogue data will also be used for training an automatic speech recognizer, which will then be integrated into the multimodal interactive system for further experimenting. Last but not least, applying reinforcement learning techniques to enhance the existing unified dialogue model centered management system is another research direction we are now concerned with.

ACKNOWLEDGMENTS We gratefully acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) through the Collaborative Research Center SFB/TR 8 Spatial Cognition – Subproject I3-SharC. We would like to thank Prof. Dr. Nicole von Steinbüchel, Frank Schafmeister and Nadine Sasse from the Department of Medical Psychology and Medical Sociology at Georg-August-Universität Göttingen for helping us to plan and execute the experiment.

REFERENCES

1. Failures diference refinement. FDR2 manual. Technical report, Formal System (Europa) Ltd., 2001.
2. J. Allen. *Natural Language Processing*. Benjamin Cumminings Publishing Company Inc., 1995.
3. J. C. Carletta. Assessing the reliability of subjective codings. *Computational Linguistics*, 22(2):249–254, 1996.
4. J. Ginzburg. Dynamics and the semantics of dialogue. *Language, Logic and Computation*, 1, 1996.
5. B. J. Grosz and C. L. Sidner. Attention, Intentions and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204, 1986.
6. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
7. S. Larsson. *Issue-Based Dialogue Management*. PhD thesis, Department of Linguistics, Göteborg University, 2002.
8. I. Lewin. A formal model of conversational game theory. In *The Fourth Workshop on the Semantics & Pragmantics of Dialogue*, 2000.
9. D. K. Lewis. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8, 1979.

10. A. W. Roscoe. *The Theory and Practice of Concurrency*. Prentice-Hall, 1998.
11. H. Shi, R. Ross, and J. Bateman. Formalising Control in Robust Spoken Dialogue Systems. In B. K. Aichernig and B. Beckert, editors, *Proceedings of Software Engineering and Formal Methods 2005*, IEEE, pages 332–341. IEEE Computer Society, 2005.
12. H. Shi, R. J. Ross, T. Tenbrink, and J. Bateman. Modelling illocutionary structure: Combining empirical studies with formal model analysis. In A. Gelbukh, editor, *Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2010)*, volume 6008 of *Lecture Notes in Computer Science*, pages 340–353, 2010.
13. S. Siegel and N. J. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. McGraw Hill, 1988.
14. S. Sitter and A. Stein. Modelling the illocutionary aspects of information-seeking dialogues. *Journal of Information Processing and Management*, 28, 1992.
15. R. Stalnaker. Assertion. *Journal of Syntax and Semantics*, 9, 1979.
16. D. Traum and S. Larsson. The information state approach to dialogue management. In *Current and New Directions in Discourse and Dialogue*. Kluwer, 2003.
17. M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella. PARADISE: A framework for evaluating spoken dialogue agents. In *Proc. of the Eighth Conference on European Chapter of ACL*, pages 271–280, 1997.
18. W. Xu, B. Xu, T. Huang, and X. Hairong. Bridging the gap between dialogue management and dialogue models. In *Proc. of the Third SIGdial Workshop on Discourse and Dialogue*, 2002.

HUI SHI
SFB/TR8 SPATIAL COGNITION,
UNIVERSITÄT BREMEN,
GERMANY
AND
SAFE AND SECURE COGNITIVE SYSTEMS,
DFKI BREMEN,
GERMANY
E-MAIL: <SHI@INFORMATIK.UNI-BREMEN.DE>

CUI JIAN
SFB/TR8 SPATIAL COGNITION,
UNIVERSITÄT BREMEN,
GERMANY
E-MAIL: <KEN@INFORMATIK.UNI-BREMEN.DE>

CARSTEN RACHUY
SFB/TR8 SPATIAL COGNITION,
UNIVERSITÄT BREMEN,
GERMANY
E-MAIL: <RACHUY@INFORMATIK.UNI-BREMEN.DE>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 175–190
RECEIVED 01/11/10 ACCEPTED 26/11/10 FINAL 15/02/11

Automatic Annotation of Referring Expressions in Situated Dialogues

NIELS SCHÜTTE, JOHN KELLEHER AND BRIAN MAC NAMEE

Dublin Institute of Technology, Ireland

ABSTRACT

To apply machine learning techniques to the production and interpretation of natural language, we need large amounts of annotated language data. Manual annotation, however, is an expensive and time consuming process since it involves human annotators looking at the data and explicitly adding information that is implicitly contained in the data, based on their judgment. This work presents an approach to automatically annotating referring expressions in situated dialogues by exploiting the interpretation of language by the participants in the dialogue. We associate instructions concerning objects in the environment with automatically detected events involving these objects and predict the referents of referring expressions in the instructions on the basis of the objects affected by the events. We judge the reliability of these predictions based on the temporal and textual distance between instruction and event. We apply our approach to an annotated corpus and evaluate the results against human annotation. The evaluation shows that the approach can be used to accurately annotate a large proportion of the utterances in the corpus dialogues and highlight those utterances for which human annotation is required, thus reducing the amount of human annotation required.

KEYWORDS: *reference resolution, situated dialogue*

1 INTRODUCTION

We present an approach to automatically annotating *referring expressions* in *situated dialogues*. A referring expression [1, Ch. 18] is an expression that occurs in natural language that is used to denote some kind of object that is discussed. For example in the sentence "Bob ate an apple", "Bob" is a referring expression that denotes some person named Bob, and "an apple" is a referring expression that denotes some apple.

The object that is being referred to is called the *referent* of the referring expression. An *anaphoric* referring expression is a referring expression that refers back to an object that has already been mentioned in the dialogue and is therefore in the linguistic context of the dialogue. An *exophoric* referring expression is a referring expression that refers to an object that has not previously been mentioned in the dialogue but that exists in some other context of the dialogue (e.g. the visual context). The process of *referring expression resolution* is the process of identifying the referents of referring expressions.

A situated dialogue is a conversation between at least two participants that takes places in an environment that is actively discussed as part of the dialogue. A typical example a of situated dialogue is a navigation task where one participant has to give instructions to a second participant to move through the environment the dialogue is situated in. Exophoric referring expressions are particularly common in this domain.

A computer system that participates in situated dialogues has to be able to resolve and produce exophoric referring expressions. There exist a number of approaches to this problem that can broadly be categorized as rule-based and machine-learning (ML) based approaches. Rule-based approaches use a number of (generally hand crafted) rules to perform the task. Grosz and Sidner [2] describe a rule-based approach to resolving reference in purely linguistic domains. Salmon-Alt and Romary present a rule-based approach to resolving reference in a multimodal domain [3].

ML based approaches on the other hand do not rely on prefabricated rules but set out to learn behaviour that is presented in the form of examples. Using ML is particularly attractive for dealing with referring expressions because using ML opens up the possibility to learn and discover strategies used by humans directly from data, which may be difficult to identify by introspection or manual analysis.

Supervised ML is a form of ML where algorithms learn a function that maps from inputs to outputs. Such algorithms require as training data a set of examples in which inputs are associated with the expected output.

Consequently, in order to train a ML algorithm to interpret or produce referring expressions, the algorithm requires a training set of examples that link spoken references to their intended referents in the world and that, furthermore, describe the conditions under which the reference was produced. These conditions may for example include the set of visible objects, the spatial relation of the speaker towards those objects and a records of previous references made by the speaker.

These training sets often have to be created manually by taking a set of inputs and annotating the expected outputs based on human judgment. This process is expensive and time consuming because it requires one or more human annotators to screen all of the examples and make a decision for each case. It is therefore desirable to find methods that can automatically perform at least parts of this process. This problem can be understood as a problem of information retrieval since the reference information must be (implicitly) contained in the data if human annotators are able to reproduce it.

Contribution: In this work we present an approach to automatically generating annotations for exophoric referring expressions in a situated task-based dialogue. We focus on identifying the referent of a referring expression, as this is a task that (unlike the determination of the set of visible objects for example), cannot be performed automatically in a straightforward manner and generally requires the attention of a human annotator. We predict the referent of a referring expression based on the interpretation of that expression in the dialogue. This is only possible if the referring expression can be related to some detectable action. We therefore only consider referring expressions in utterances that instruct the hearer to perform some specific task. In the experiments described in this work we focus on one specific kind of instruction, namely instructions to pass through a door.

Overview: In Section 2 we discuss corpora that are possible fields of application for our approach and introduce the corpus that is used in the example presented in this work. In Section 3 we present our approach to detecting the referents of referring expressions. In Section 4 we present the evaluation of the application of our approach to test data. Finally, in Section 5 we discuss these results and possible extensions of this work.

2 DATA

For this experiment we were interested in corpora featuring situated dialogue. In addition to information immediately related to the dialogue,

such as transcriptions and annotations, we were also interested in additional data related to the environment, such as maps and recordings of the actions of the participants.

There exist a number of freely available situated dialogue corpora. The TRAINS corpus [4], which contains dialogues between two participants planning train routes on a map, is an example of a corpus that incorporates the visual modality, and has transcriptions, but does not feature reference annotations. In addition to that, the corpus works with a static map, which is not dynamically updated, which makes it difficult to annotate referring expressions, because participants frequently talk about hypothetical scenarios. In addition to this, it also lacks a record of the planned routes.

Another visually situated corpus is the MAPTASK corpus [5]. This corpus is based on an experiment where one participant describes a route in a map to a second participant, who has access to a slightly different map. Navigation takes place at an abstract level which makes it hard to identify events at a level that would be relevant to this experiment.

The corpus considered in this work is the SCARE corpus [6]. This corpus consists of dialogues between two participants in a navigation task where the environment is perceived from a first person perspective. It contains transcriptions and reference annotations and is therefore a good example for learning referring expression resolution. Moreover, unlike the TRAINS and MAPTASK corpus, the SCARE corpus features recordings of all navigation steps, thereby enabling us to reconstruct actions performed by the player.

Fig. 1. Screenshot of a video recording from the SCARE corpus.

What differentiates the SCARE corpus and makes it particularly interesting to us, is that it does not take a remote approach with an external perspective, but is very situated, by putting the participants inside the environment. This means that the participants have a location in the environment, which restricts references and actions thereby creating the possibility of linking them. This is the key to our approach. The corpus was created in an experiment focusing on situated task-based dialogues. In this experiment one participant, the direction follower (DF), had to navigate through an environment simulated in a game engine, while the second participant, the direction giver (DG), had to give directions to the first participant to help them fulfil a given task. The details of the task and the layout of the world were known only to the DG. The DF navigated through the environment in a first person perspective, of which a live video feed was shown to the DG. Therefore both participants had the same perspective on the environment. The participants communicated through a voice connection.

The corpus comprised video and audio recordings of the dialogues, as well as transcriptions of the audio files that were annotated for reference, i.e. referring expressions that referred to objects in the environment were annotated with which object the expression referred to. In addition to that, demo files were provided that could be replayed in the game engine, thereby recreating the navigation movements in each dialogue.

3 EXTENDING THE SCARE CORPUS

As noted in Section 2, the SCARE corpus contains annotated dialogue transcriptions and a record of player movement. In order to automatically annotate referring expressions, we needed to create new data from the corpus. In particular, we had to identify a set of referring expressions and then determine the referent for each expression. We did this by establishing a correspondence between instructions that contain a referring expression in the dialogue and events in the world that could be caused by these instructions. The events we wanted to consider were not explicitly contained in the data, so we had to reconstruct them. Consequently, establishing a correspondence between instructions in the dialogue and events in the world involved 3 steps:

1. We detected a set of instructions.
2. We detected a set of events.

3. We established a correspondence between instructions and events and recorded values for different distance metrics between instructions and events.

Each of these steps is described in detail below. We then evaluated the correspondence against gold standard manual annotations. This evaluation is described in Section 4.

3.1 *Detecting the Instructions*

In this experiment we were interested in referring expressions that caused events we could detect by looking at the movement of the player in the environment. One class of such events is passing through doors. We therefore detected instances of the DG telling the DF to go through a door. We did this using a regular expression of this form:

```
[go|pass]through.*[door|one|that]
```
[1]

This expression fit instructions such as "go through the right door" or "pass through the next one". We collected instructions up to a length of seven words. The regular expression was defined by examining a small number of the dialogues in the SCARE corpus. In total we detected 135 referring expressions using this regular expression. This approach probably did not capture all instructions, but served as a good starting point.

3.2 *Detecting Events*

Once we had detected the set of instructions that we would use in our experiment we then had to detect the set of relevant events to match against the instructions. To do this, we replayed the demo files in the game engine and recorded the position and orientation of the player during the dialogues. We then aligned this information with time. By comparing this information with geometric information about the layout of the rooms, we were able to detect the moments when the player left a room and entered another room. This in turn enabled us to determine which door the player had passed at what point in time. Each passing of a door formed an event.

[1] $.*$ matches any sequence of characters, $[x|y]$ matches the sequence x or y.

3.3 *Establishing the Correspondence*

In this step we determined a correspondence between instructions and events for our example corpus. We aimed to identify for each instruction the specific event that occurred when the DF fulfilled the instruction. Events naturally occur slightly after the instruction has been produced because the DF needs time to interpret the instruction and to navigate into a position where it is possible to perform the required action. However, not every instruction is immediately succeeded by an event that fulfils the instruction. We see three main reasons for this:

1. The DF may misinterpret the instruction and begin to perform a different action.
2. The DF may not understand an instruction or find it ambiguous and ask the DG to clarify. In this case, the next event may follow after a longer delay, during which the participants come to an agreement about the next action, and may actually end up not fulfilling the original instruction because the participants decided on a different course of action.
3. A number of other events may occur between an instruction and the corresponding event because the DF has to fulfil a number of subgoals in order to be able to fulfil the instruction.

At first glance, two approaches in creating a correspondence are apparent: we can either start out with the events and search for an instruction to match each event; or we can start out with the instructions, and determine which event was caused by each instruction. The first approach immediately appeared less favourable because in the example dialogues, a great number of events are not directly caused by instructions. This happens when the DF is exploring the map on their own, or if the DG gives high level goals, such as returning to a previously visited room, which the DF can fulfil without being instructed in every step. We therefore decided to use the approach where we start with the instructions and then search for events that match these instructions.

We processed each dialogue incrementally by going through it from the beginning, picking up instructions and events as they occurred. An incoming instruction was processed by storing it on a FIFO queue. An incoming event was processed by removing the oldest instruction from the queue and associating it with the new event, and storing the resulting pair for later evaluation. This was based on the assumption that events were preceded by instructions. Roughly speaking this approach associates each instruction with the next event occurring after it.

We collected instructions in a queue, which enabled us to correctly interpret concatenated instructions (e.g. “go through this door and then go through the next one”) as two instructions to be executed sequentially.

Events that occurred while the instruction queue was empty were discarded as events that occurred without explicit instruction. Such events occurred often in the dialogues when the DF was asked to move to a previously visited location. In this case, the DF often could find the way on their own without having to be explicitly instructed for each step.

The matching process resulted in a set of pairs of instructions and events which was the basis for our experimental evaluation. The algorithm we used for this process is presented as Algorithm 1.

Algorithm 1 The algorithm for associating instructions and events.

Input:	**dialogue:** a temporally ordered set of events and instructions.
Data Structures:	**instructionQueue:** a Queue for incoming instructions, initially empty.
	correspondences: a list of instructions-event pairs.
Output:	a list of instructions-event pairs which may be related.

```
FOR the length of the dialogue
   e := select the next event or instruction
   IF e is an instruction
      push(e, instructionQueue)
   ELSE
      IF e is an event
         IF empty(instructionQueue)
            discard e
         ELSE
            i := pop(instructionQueue)
            a := associate(i,e)
            append(a, correspondences)
         ENDIF
      ENDIF
   ENDIF
ENDFOR
RETURN correspondences
```

Every time an instruction and an event were associated, we recorded the distance in time between instruction and event, and the number of

words spoken between them to facilitate evaluation. We derived these values from the time aligned dialogue transcriptions.

The output of the algorithm consists of a list of associated instructions and events. Each pair represents a possible causal relationship between an instruction and an event, and thereby a candidate for annotation. In the next step of the process each pair will be more closely examined, and it will be estimated how likely the pairing is to be a correct assignment.

Figure 2 illustrates the approach. Intervals of speech are represented as blocks below the time axis. Dark blocks represent instructions, while bright block represent speech that is not an instruction. Stars on the time axis represent events. The horizontal brackets delineate the intervals between the end of an instruction and the next event. The dashed vertical lines cut out intervals on the time axis and pieces of the speech blocks, which form the distance values.

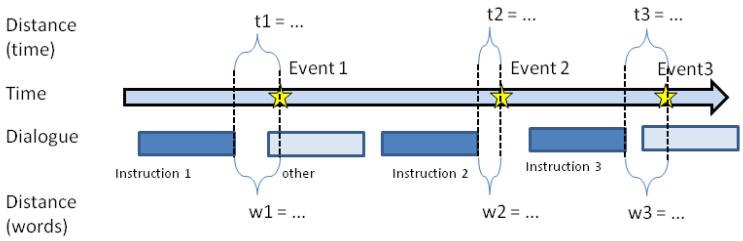


Fig. 2. Illustration of the instruction-event association and distance measuring process. Blocks represent intervals of speech, stars represent events.

4 EVALUATION

As mentioned in Section 2, referring expressions in the original corpus were annotated for reference. We therefore knew for each referring expression to which object it actually referred. This information formed the gold standard for the evaluation of our approach to reference resolution.

Once we had processed all the dialogues in the corpus we started the evaluation. As the first step we defined the baseline for the evaluation. We did this by taking the unmodified instruction/event pairs. In this set, each instruction was associated with the closest following event. This

is a relatively simple way of associating instructions and events since it assumes that each instruction was perfectly interpreted and fulfilled directly after the instruction, with no other events occurring between them. This is a very strong assumption, because misunderstandings between human communicators frequently occur. This results in the user executing a wrong action or not immediately performing the action. We therefore suspected that this initial association contained many false pairings.

We take this set of associations as the baseline in this experiment in the sense that this association is the most simple but plausible one that can be created without much effort.[2]

We then set out to detect likely false pairings by looking at the distance between instruction and event.

We used two basic approaches: If the distance between an instruction and the following event exceeded a given threshold, we would refuse to rate it, leaving the decision up to a human annotator ("late cut-off"). If the distance fell below a given threshold, we also did not rate the pair ("early cut-off"). In an actual annotation scenario, the examples that were not rated could be passed on to a human annotator who could judge them manually.

We ran the association algorithm (Algorithm 1) to create a set of instruction-event pairs. We subsequently judged the results by a number of different distances. The results for the time distances are presented in Table 1 , the results for word distances in Table 2. They show:

- the total number of cases (Column 1)
- the number of cases that were removed because of the cut-off criterion (Column 2)
- the percentage of removed cases (Column 3)
- the number of remaining cases (Column 4)
- the number of cases where the association between instruction and event was correct according to the gold standard (Column 5). This row is illustrated in Figure 3(a) for time distances and Figure 3(b) for word distances.
- the percentage of correct cases among the cases that were not removed (Column 6)

[2] A different measure that could serve as a basis for evaluating results later on would be the stochastic probability of picking the right referent when choosing randomly among the visible objects. In a related experiment [7] we determined this probability to be 57.4% for this corpus. However, in the current experiment we cannot assume that the intended referent is visible, therefore this approach is not directly applicable.

– the overall percentage of the correctly associated cases among the number of total cases (Column 7)

For early cut-off, we used the distances 5, 7.5, 10, 15 and 20 seconds and 5, 10, 20, 40, 50 and 60 words. The results are displayed in Table 3 and 4.

Table 1. Results for the different time distance values for late cut-off.

	Total	Removed		Remaining	Correct		Total correct
	#	#	%	#	#	%	%
Col. Nr.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Baseline	135	0	0.0	135	93	68.9	68.9
Time 5s	135	88	65.2	47	34	72.3	25.2
Time 7.5s	135	61	45.2	74	61	82.4.	45.2
Time 10s	135	40	29.6	95	80	84.2	59.3
Time 15s	135	34	25.2	101	84	83.2	62.2
Time 20s	135	26	19.3	109	88	80.7	65.2

Table 2. Results for the different word distance values for late cut-off.

	Total	Removed		Remaining	Correct		Total correct
	#	#	%	#	#	%	%
Col. Nr.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Baseline	135	0	0.0	135	93	68.9	68.9
Words 5	135	122	90.4	13	4	30.8	2.9
Words 10	135	102	75.6	33	21	63.6	15.5
Words 20	135	62	45.9	73	59	80.8	43.7
Words 40	135	38	28.1	97	80	82.5	59.3
Words 50	135	31	23.0	104	85	81.7	63.0
Words 60	135	26	19.3	109	86	78.9	63.7

To give an intuition about the significance of the different columns: Column (3) tells us for what fraction of the cases the algorithm refused to make a judgment. The figure basically tells us how much work is left for the human annotator. Column (6) tells us how many of the cases that were not removed were actually correct. This basically gives us a measure of the quality of the predictions made.

Table 3. Results for the different time distance values for early cut-off.

	Total	Removed		Remaining	Correct		Total correct
	#	#	%	#	#	%	%
Col. Nr.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Baseline 0	135	0	0.0	135	93	68.9	68.9
Time 1s	135	7	5.2	128	93	72.7	68.9
Time 2s	135	8	5.9	127	93	73.2	68.9
Time 2.5s	135	13	9.6	122	89	73.0	65.9
Time 5s	135	47	34.8	88	59	67.0	43.7

Table 4. Results for the different word distance values for early cut-off.

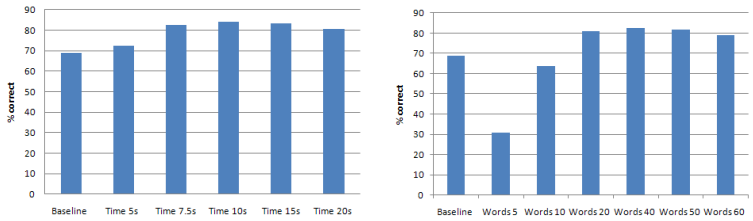
	Total	Removed		Remaining	Correct		Total correct
	#	#	%	#	#	%	%
Col. Nr.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Baseline	135	0	0.0	135	93	68.9	68.9
Words 1	135	3	2.2	132	93	70.5	68.9
Words 2	135	4	2.9	131	93	71.0	68.9
Words 3	135	6	4.4	129	93	72.0	68.9
Words 4	135	7	5.2	128	92	71.9	68.1
Words 5	135	11	8.1	124	90	72.6	66.7
Words 7	135	16	11.9	119	86	72.3	63.7

Column (7) tells us which fraction of the total number of cases was correctly annotated according to the manual annotations from the corpus.

As we can see, the baseline alone delivers somewhat acceptable results. However, if we were to use the baseline approach in an actual annotation task, we would end up with false results with no indication of which results were doubtful decisions.

Using the cut-off approach removes cases while increasing the correctness of the remaining ones. This means that the cut-off strategy helps us identify cases that are likely to be incorrect.

For late cut-off we observe that low threshold values remove many cases while higher values remove less cases. We also observe that that the overall correctness of the remaining cases peaks at a certain point (around 10 words for the word distance cut-off and 40 seconds for the time distance cut-off) and decreases for greater values. This may seem counter-intuitive, but can be explained: early cut-off values remove the majority of cases, including many correct ones, and tend to preserve cases



(a) Proportion of correct judgements by time distance.

(b) Proportion of correct judgements by word distance.

Fig. 3. Graphs showing the distribution of correct judgements for late cut-off.

where instruction and event are very close together. As discussed earlier, it is a reasonable assumption that these cases tend to be incorrect matches.

The observations indicate that this is indeed the case and highlights the need for trying out the early cut-off approach.

In early cut-off, small values remove few cases and large values remove many. Again we observe a peak and subsequent drop in correctness. Early cut-off achieves at best a correctness around 73% while late cut-off achieves a correctness around 83%, which in both cases is a clear improvement over the baseline.

The results indicate that both early and late cut-off remove incorrect candidates, thereby increasing the correctness of the remaining candidate set. In addition to that we know early and late cut-off remove cases from opposing sides of the spectrum (cases where instruction and event are close together and cases where the opposite is the case). It is therefore likely that one approach captures cases the other does not cover. It is therefore promising to develop an approach that integrates both.

5 Conclusions and Future Work

We presented an approach towards automatically generating referring expression annotations for situated dialogues that exploits the interpretation of referring expressions by the participants of the dialogue. We demonstrated the approach for a specific type of references in a specific corpus. The approach can be generalized to other types of references in other corpora under two conditions: (1) The references must be contained in instructions that cause events involving the referents and (2) It must be possible to automatically detect these events.

On a conceptual level we can relate this approach to more general approaches that are based on intention recognition and perceived affordances [8]. In [9] Gorniak et. al. describe using intention recognition to improve reference resolution in the context of a game. In this work we somewhat reverse this approach: We take actions in the game as hypotheses about the intention of instructions (quasi hijacking the interpretation performed by the listener) and use the objects affected by the action as the referent of referring expressions in the instruction.

We explored different early and late cut-off values that give an indication for which suggested linkings might be unreliable. Deciding on a particular cut-off point, allows the algorithm to decide which cases are easy and reliably judged, and which cases are hard to judge, and should rather be inspected by a human annotator. However, it is not immediately clear how to derive cut-off values for new domains. It may be possible to directly transfer values between sufficiently similar domains. Another approach would be to manually create a gold standard annotation for a small subset of the domain and to determine values for this subset and transfer them to the whole domain.

Overall, the approach manages to produce at best a success rate around 80% if only one cut-off strategy is used.

To increase this value, we are investigating the use of cut-off windows instead of cut-off points. The results of the experiments suggest that very early events as well as very late events are poor candidates for annotation. Therefore it appears to be sensible to remove early as well as late events. On a trial basis we combined different good values for early and late cut-off and achieved a success rate around 93%, while still annotating about 45% of all cases (due to lack of space we unfortunately cannot present this data). While this is still quite a bit away from correctly annotating all examples, it still enables us to automatically annotate a sizeable subset of cases with good success rate.

The GIVE corpus [10] comprises a data set, that is very similar to the one we used, but is based on written instead of spoken language and features only monologue. In further work we may investigate how well our approach can be applied to this corpus and in how far results are transferable.

REFERENCES

1. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech

Recognition. second edition edn. Prentice Hall (2008)
2. Grosz, B.J., Weinstein, S., Joshi, A.K.: Centering: A framework for modeling the local coherence of discourse. Computational Linguistics **21**(2) (1995) 203–225
3. Salmon-Alt, S., Romary, L.: Reference resolution within the framework of cognitive grammar. In: Proceedings of the International Colloqium on Cognitive Science. Volume abs/0909.2626., San Sebastian, Spain (2000)
4. Heeman, P., Allen, J.: Trains 93 dialogues. Technical report, University of Rochester, Rochester, NY, USA (1995)
5. Thompson, H., Anderson, A., Bard, E.G., Doherty-Sneddon, G., Newlands, A., Sotillo, C.: The HCRC Map Task corpus: Natural dialogue for speech recognition. In: Proceedings of the workshop on Human Language Technology, HLT-93, Princeton, New Jersey, Association for Computational Linguistics (1993)
6. Stoia, L., Shockley, D.M., Byron, D.K., Fosler-Lussier, E.: Scare: A situated corpus with annotated referring expressions. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008). (2008)
7. Schütte, N., Kelleher, J., Mac Namee, B.: Visual salience and reference resolution in situated dialogues: A corpus-based evaluation. In: Dialog With Robots. Papers from the AAAI Fall Symposium, Menlo Park, California, AAAI Press (2010)
8. Gorniak, P.: The Affordance-Based Concept. Ph.d., Massachusetts Institute of Technology (2005)
9. Gorniak, P., Orkin, J., Roy, D.: Speech, space and purpose: Situated language understanding in computer games. In: Twenty-eighth Annual Meeting of the Cognitive Science Society Workshop on Computer Games. (2006)
10. Gargett, A., Garoufi, K., Koller, A., , Striegnitz, K.: The Give-2 corpus of giving instructions in virtual environments. In: Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC), Valletta, Malta, European Language Resources Association (ELRA) (2010)

NIELS SCHÜTTE
APPLIED INTELLIGENCE RESEARCH CENTRE,
DUBLIN INSTITUTE OF TECHNOLOGY,
IRELAND
E-MAIL: <NIELS.SCHUTTE@STUDENT.DIT.IE>

JOHN KELLEHER
APPLIED INTELLIGENCE RESEARCH CENTRE,
DUBLIN INSTITUTE OF TECHNOLOGY,
IRELAND
E-MAIL: <JOHN.D.KELLEHER@DIT.IE>

BRIAN MAC NAMEE
APPLIED INTELLIGENCE RESEARCH CENTRE,
DUBLIN INSTITUTE OF TECHNOLOGY,
IRELAND
E-MAIL: <BRIAN.MACNAMEE@DIT.IE>

Machine Translation and Multilingualism

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 193–208
RECEIVED 23/10/10 ACCEPTED 26/11/10 FINAL 08/02/11

Phrase-based Machine Translation based on Text Mining and Statistical Language Modeling Techniques

CHIRAZ LATIRI,[1] KAMEL SMAÏLI,[2] CAROLINE LAVECCHIA,[2]
CYRINE NASRI,[1] AND DAVID LANGLOIS[2]

[1] *El Manar University, Tunisia*
[2] *LORIA, France*

ABSTRACT

In this paper, we introduce two new methods dedicated to phrase-based machine translation. Both are based on mining a parallel corpus in order to find out the couples of linguistic units which are translation of each other. The presented methods do not rely on any alignment in contrast to what is done usually by the statistical machine translation community. Each of them proposes a complete translation table containing translations of single words and phrases. The first method is inspired from the well-known trigger language model while the second one is inspired from the association rules mining technique. All experiments are conducted on a large part of EUROPARL *corpus and highlight the utility of both proposed approaches.*

KEYWORDS: *Statistical machine translation, Sequence mining, Inter-lingual triggers, Inter-lingual association rules, Bilingual corpora.*

1 INTRODUCTION

Since the apparition of the pioneering work of IBM researchers [1], almost all the proposed papers in Statistical Machine Translation (SMT) are based on their formalism. This is due to the strength of the approach

and the availability of tools, such as GIZA++ for producing the translation table [2], CMU or SRILM for developing language model [3] and PHARAO [4] or MOSES [5] for decoding. These tools make developing SMT a very easy process.

In this paper, we would like to show that it is possible to investigate other issues which could constitute an alternative to IBM methods and their generalization to support phrase-based models [6]. The proposed methods do not rely on any alignment in contrast to what is done usually by the SMT community. The first method is inspired from the well-known trigger language model which we adapted to automatically learn words and phrases equivalents from bilingual corpora. The second one is inspired from the association rules mining technique, well-known in data mining. We adapt this latter to make it supporting two different languages.

The paper is organized as follows: Section 2 recalls the basic foundations of statistical machine translation. We devote Section 3 to present the machine translation approach based on inter-lingual triggers. Section 4 introduces the second one based on inter-lingual association rules. Then, in Section 5, we present results of the mixture of the two above methods. The conclusion and future works are presented in Section 6.

2 PRINCIPLE OF STATISTICAL MACHINE TRANSLATION

In SMT framework, the translation process comes back essentially to the search for the most probable sentence f in the target language given a sentence e in the source language. Let $e = e_1, .., e_j$ be the source sentence (*i.e.*, to be translated) and $f = f_1, .., f_i$ be the sentence generated by the translation system, namely:

$$\hat{f} = \arg\max_{f} P(f|e) \tag{1}$$

By using the Bayes formula, we obtain:

$$\hat{f} = \arg\max_{f} P(f)P(e|f) \tag{2}$$

In Equation (2), $P(f)$ is estimated by a *language model*. Its role is to propose a sentence supposed to be correct in the target language. $P(e|f)$ is computed from a *translation model* and is supposed to reflect the truthfulness of the translation. Then, the decoder like PHARAO [4] or MOSES [5] generates the best hypothesis by making a compromise between, at least, these probability distributions.

3 STATISTICAL MACHINE TRANSLATION USING INTER-LINGUAL TRIGGERS

The concept of *triggers* has been largely used in statistical language modeling [7, 8]. Roughly speaking, a statistical language model yields a probability to each potential sequence of words belonging to a vocabulary. A trigger model enhances the probability of a list of words which are correlated to a word w_i. To develop such a model, all the correlated words are retrieved. Triggers are determined by computing *mutual information* between two linguistic units x and y, each of them takes its values in the list of words belonging to the vocabulary V. Given two words x, y, the correlation $\mathrm{MI}(x, y)$ is given by:

$$\mathrm{MI}(x, y) = P(x, y) \log_2 \frac{P(x, y)}{P(x)P(y)} \tag{3}$$

$P(x, y)$ is the joint probability between x and y, while $P(x)$ and $P(y)$ are the marginal probabilities of x and y, respectively.

3.1 *Inter-lingual triggers*

In [9], the concept of triggers is adapted to handle relationships between words for any two different languages. This approach is called *inter-lingual triggers*. An inter-lingual trigger is henceforth a set composed of a word (or a phrase) f in a source language, and its corresponding best correlated words (or phrases) in a target language $e_1, e_2, \ldots, e_n$. This will be written as:

$$\mathrm{Trig}(f) \longrightarrow e_1, e_2, \ldots, e_n \tag{4}$$

In inter-lingual triggers f takes its values from the source vocabulary (French) and e_i from the target one (English). The translation table is obtained by assigning to each inter-lingual trigger a probability calculated as follows:

$$\forall\, f, e_i \in \mathrm{Trig}(f), P(e_i|f) = \frac{\mathrm{MI}(e_i, f)}{\sum\limits_{e_j \in \mathrm{Trig}(f)} \mathrm{MI}(e_j, f)} \tag{5}$$

where $\mathrm{Trig}(f)$ is the set of k English linguistic units triggered by the French unit f.

In [10, 9], a word-based machine translation using inter-lingual triggers is detailed and results are presented. This approach is extended in the following to achieve a phrase-based machine translation approach.

3.2 *A new approach to achieve phrase-based MT*

Since more than ten years, researches showed that the use of phrases in translation instead of words leads to better SMT system quality. In order to retrieve phrases, several approaches have been proposed in the literature. Most of them require word-based alignments. For instance, Och *et al.* in [11] collect all phrase pairs that are consistent with the word alignment provided by Brown's models.

In this paper, we will show how to take advantage from inter-lingual triggers and how to make machine translation supporting phrases generated by triggers without any alignment. A sequence of n French words can trigger a sequence of m English words with $n, m \in \mathbb{N}$. This kind of correlation is denoted by $n.m$-Trigger. In the remainder, we will detail the different steps of our approach.

PHRASE EXTRACTION To retrieve from a corpus pertinent phrases, we use a method developed in [12], to rewrite the source training corpus in terms of phrases. To achieve that, an iterative process selects phrases by grouping words which have a high Mutual Information value. Only phrases improving the perplexity are kept for the forthcoming steps. At the end of the process, we get a list of phrases and a source corpus rewritten in terms of these discovered phrases. With this source corpus expressed with pertinent phrases, we aim to find their potential phrase translations in the target corpus by using inter-lingual triggers.

LEARNING PHRASES TRANSLATIONS Since our method does not require any alignment, we assume that each source phrase of l words could be translated by several sequences of $l \pm \Delta l$ words. This means, to each source phrase, we associate $(2\Delta l + 1)$ sets of its k best inter-lingual triggers. Each set S_i is composed of the potential translations of i words with $i \in [l - \Delta l, .., l + \Delta l]$ with $l - \Delta l \geq 1$. Thus, we allow a source phrase to be translated by different target sequences of variable sizes. Table 1 shows the potential translations of the source phrase *Porter plainte*.

For the cited example, we guess that Δl is set to 1. Consequently, *"porter plainte"* could be translated by a sequence of at least one word and at most by a sequence of 3 words. In this example, we have selected 9 potential translations. Obviously, only *"press charges"* is correct. In the general case, each phrase could be translated by k potential units. That is why we propose to select those which are pertinent and discard the noisy ones.

Table 1. Potential translations of the source phrase *"porter plainte"*.

	$n.m$-**Triggers**		
Source phrase	2.1	2.2	2.3
porter plainte	press	press charges	can press charges
	charges	can press	not press charges
	easy	not press	you can press

Algorithm 1: Simulated Annealing (SA) algorithm.

begin
2 Start with a high temperature T;
3 **repeat**
4 From the current temperature T, state i and a BLEU B_i, randomly add a subset of $n.m$-Triggers into the translation table which makes the system moving from state i to j. With this new table, we run a decoder. This leads to different hypotheses which are evaluated using BLEU on the development corpus. We get a new BLEU B_j.
5 **if** $B_j - B_i >= 0$ **then**
6 state j is kept as the new current state
7 **else**
8 j is accepted as the new current state with a probability $random(P) < e^{\frac{B_i - B_j}{T}}$ with $P \in [0 \dots 1]$
9 **until** *BLEU equilibrium with temperature T is reached*;
10 Decrease the temperature and go to line 3 until the given low temperature is reached or until the BLEU stops increasing.

All source phrases and their sets of inter-lingual triggers constitute the set of $n.m$-Triggers. The main challenge is how to select the best $n.m$-Triggers. In other words, what are the pertinent phrases and their translations. The choice of the best sub-set phrases is a combinatorial problem. Simulated annealing (SA) algorithm is one of the algorithms which can give a good solution to this kind of problem. We have yet used SA in previous work [13] for automatic word clustering. Basing on this experience, we decided to choose this algorithm among all possible ones to solve our problem. To achieve that, we start with a word-based MT system based on 1.1-Triggers presented in [10]. Then, we randomly add phrases ($n.m$-Triggers) into the translation table until an optimal BLEU score is reached on a development corpus. In other words, we only keep

phrases which improve the quality of translation on a development corpus. This method is summarized in Algorithm 1.

For each French unit (a word or a sequence) of l words, we select from the target training corpus its $10 \times (l \pm \Delta l)$ best inter-lingual triggers (translations). This means that if a sequence of l words is allowed to be translated by $l-2$, $l-1$, l, $l+1$, $l+2$ words, then 50 potential candidates are kept. All this inter-lingual triggers make the set of candidate phrase translations (called $n.m$-Triggers) required by the SA algorithm.

In the next section, we present another original method which uses association rules between terms in SMT.

4 MACHINE TRANSLATION WITH INTER-LINGUAL ASSOCIATION RULES

The association rules mining problem has been introduced by Agrawal *et al*. [14]. The motivation for searching associations from texts is to discover correlations between terms that occur together as well as to look for regularities in corpora. Before presenting our method, let us give a brief review of the basic definitions related to association rule mining [14].

Definition 1. *An extraction context (or* corpus*, in our case) is a triplet* $\mathcal{K} = (\mathcal{P}, \mathcal{T}, \mathcal{R})$ *where:*

- $\mathcal{P} = \{s_1, s_2, \ldots, s_n\}$ *is a finite set of* n *distinct sentences of a corpus.*
- $\mathcal{T} = \{t_1, t_2, \cdots, t_m\}$ *is a finite set of* m *distinct terms of a corpus.*
- *Both sets* $\mathcal{T}$ *and* $\mathcal{P}$ *are linked through a binary relation* $\mathcal{R}$ *such that* $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{T}$*. That is, each sentence* $s \in \mathcal{P}$ *is represented a set of terms* m *terms* $T \in \mathcal{T}$ *named termset, that occur together in the sentence.*

The support of $X \subseteq \mathcal{T}$ in $\mathcal{K}$, denoted by $Supp(X)$, is the absolute number of a randomly chosen sentences from $\mathcal{P}$ containing the termset X. A k-termset $T \in \mathcal{T}$, *i.e.,* a termset of length k, is called *frequent* if the k terms of T occur simultaneously in the corpus more than a user-defined frequency threshold denoted *minsupp*.

Definition 2. *An association rule* R *over* $\mathcal{K}$ *is an implication of the form* $R : X \Rightarrow Y$*, where* X *and* Y *are subsets of* $\mathcal{T}$*, and* $X \cap Y = \emptyset$*. The termsets* X *and* Y *are, respectively, called the premise and the conclusion parts of* R*.*

The *support* of a rule R and its *confidence* are defined as:

$$Supp(R) = Supp(X \cup Y) \qquad Conf(R) = \frac{Supp(X \cup Y)}{Supp(X)} \tag{6}$$

An association rule R is *valid* if its confidence value is greater than or equal to a user-defined threshold, denoted *minconf*.

For word-based machine translation, we introduced in [10] the concept of *Inter-lingual association rules*, named ILAR. The potential translations of a French term f are obtained by selecting all the English terms $e_1, e_2, \ldots, e_n$ which are present in the conclusion of a inter-lingual association rule for which f is its premise.

Since we are interested in phrase-based machine translation, we investigate in the following the problem of mining frequent closed sequences from highly sized bilingual corpora. Our aim is to extend the concept of inter-lingual association rules to the context of phrase-based machine translation.

4.1 *Mining frequent closed sequences for phrase-based machine translation*

Our approach is inspired from an efficient sequential pattern mining algorithm, called BFSM [15]. Our choice of BFSM is argued by the fact that this latter is well adapted for handling very large corpora and, especially for low values of the support threshold. A set of frequent closed sequences is retrieved and then inter-lingual association rules are obtained from this latter as explained further.

We consider the context $\mathcal{K} = (\mathcal{P}, \mathcal{T}, \mathcal{R})$ (*cf.* Definition 1).

Definition 3. *A sequence $S = \langle t_1, \ldots, t_j, \ldots, t_n \rangle$, such that $t_k \in \mathcal{T}$ and n is its length, is a n-termset for which the position of each term in the sentence is maintained. S is called a n-sequence.*

Definition 4. *A sequence $S_\alpha = \langle a_1, a_2, \ldots, a_n \rangle$ is a sub-sequence of $S_\beta = \langle b_1, b_2, \ldots, b_m \rangle$, denoted by $S_\alpha \subseteq S_\beta$, if there is a set of indices $1 \leq i_1 \leq i_2 \leq \ldots \leq i_n \leq m$, such that $a_1 = b_{i_1}, a_2 = b_{i_2}, \ldots, a_n = b_{i_n}$. S_β is called a* super-sequence *of S_α.*

Definition 5. *Given S a sequence discovered from $\mathcal{K}$. The support of S is the number of sentences in $\mathcal{P}$ that contain S,* i.e., *$Supp(S) = \| \; p \in \mathcal{P}$ s.t. $S \subseteq p \; \|$. S is said to be* frequent *if and only if its support is greater than or equal to the minimum support threshold $minsupp$.*

Definition 6. *A frequent closed sequence (FCS) S is a frequent sequence that has no frequent super-sequence $S^{'}$ with the same support.*

Definition 7. *Given a sequence S, its information position, denoted by $\mathcal{POS}_S$, is a set of pairs (id_s, pos_seq), where id_s represents the rank of the sentence in the corpus and pos_seq the position of the last term of the sequence in the sentence.*

Table 2. Example of a sequences dataset.

Sentence	Sequence
s_1	$S_1 : \langle text, mining, tools, for, text \rangle$
s_2	$S_2 : \langle text, mining, for, analysis \rangle$
s_3	$S_3 : \langle text, for, analysis \rangle$
s_4	$S_4 : \langle text, mining, for, analysis \rangle$

To illustrate this concept, let us take an example. Given a dataset of sequences depicted in Table 2 and a value of the $minsupp$ threshold equals to 2. We can get the information positions of the frequent 1-sequences as shown in Table 3. We notice that the term $tools$ is pruned since its support is lower than the $minsupp$ value.

Table 3. The frequent 1-sequences.

1-Sequence	Information position	Support
$\langle text \rangle$	(1 , 1) (1 , 5) (2 , 1) (3 , 1) (4 , 1)	4
$\langle mining \rangle$	(1 , 2) (2 , 2) (4 , 2)	3
$\langle for \rangle$	(1 , 4) (2 , 3) (3 , 2) (4 , 3)	4
$\langle analysis \rangle$	(2 , 4) (3 , 3) (4 , 4)	3

Proposition 1. *Given two sequences S_k and $S_{(k+n)}$. Then if $S_{(k+n)}$ is a super-sequence of S_k and if they have the same k first terms and the same support, then $S_{(k+n)}$ is called* ***a backward super-sequence*** *of S_k* [15].

For instance, the sequence ⟨*Statistical machine translation evaluation*⟩ is a backward super-sequence of the sequence ⟨*Statistical machine translation*⟩ since they share the three first terms, while assuming that they have the same support. The second sequence is then considered redundant.

Thus, the set of the frequent closed sequences, denoted by $\mathcal{FCS}$ only contains non-redundant patterns, *i.e.*, those not covered by other ones of the same support (or equivalently, do not have a backward super-sequence, *cf.* Proposition 1).

AN ALGORITHM FOR DISCOVERING FREQUENT CLOSED SEQUENCES The main idea of frequent closed sequence mining is based on the principle of *sequence-extension* [15]. The sequence-extension of a k-sequence S_k adds a new term to S_k as a new last element. The frequent $(k+1)$-sequences can be produced by extending the current found frequent k-sequences. Indeed, for each frequent k-sequence S_k, we pick up each frequent 2-sequence S_α whose first term is the same as the last term of S_k and matches the information position of S_k. The result is a new frequent $(k+1)$-sequence. Algorithm 2 details how extending a frequent k-sequence by a frequent 2-sequence according to the explained process.

In the algorithm, $\oplus$ denotes the concatenation operator.

Algorithm 2: Sequence-Extension.

Input: a k-sequence S_k and a 2-sequence S_α.
Output: a $(k+1)$-sequence
begin
  if *last-term*(S_k) = *first-term*(S_α) $\wedge$ $id\_s(S_k) = id\_s(S_\alpha)$
    $\wedge$ $pos\_seq(S_\alpha) = pos\_seq(S_k) + 1$ **then**
    $S_{k+1} = \langle S_k \oplus (S_\alpha \setminus$ *first-term*$(S_\alpha))\rangle$;
  return S_{k+1};

Our approach proceeds in four steps, namely:

Step 1: Extraction of frequent 1-sequences and their information positions. We scan the extracted context $\mathcal{K}$ once to record the information position (*cf.* Definition 7) of each distinct term in the corpus.

Step 2: Generation of the frequent 2-sequences. The frequent 2-sequences are produced by applying join operation on the 1-sequences as in the APRIORI-like methods [14]. Note that, we do not need to scan the corpus to count their supports. Indeed, the information positions of the frequent 1-sequences are used to get those of frequent 2-sequences.

Let us consider the two 1-sequences $\langle text\rangle$ and $\langle mining\rangle$ given in Table 3. The 2-sequence $\langle text\ mining\rangle$ is generated as follows: the pair $(1,2)$ of $\langle mining\rangle$ is matched with the pair $(1,1)$ of $\langle text\rangle$. By the same way, the pairs $(2,2)$ and $(2,1)$ are matched to form the pair $(2,2)$, and the pairs $(4,2)$ and

$(4, 1)$ to form the pair $(4, 2)$. So, the information position of the 2-sequence $\langle text\ mining \rangle$ are the pairs $(1, 2)$, $(2, 2)$ and $(4, 2)$ (*cf.* Table 4).

Table 4. The frequent 2-sequences.

Frequent 2-sequence	Information position
$\langle text\ mining \rangle$	(1 , 2) (2 , 2) (4 , 2)
$\langle text\ for \rangle$	(1 , 4) (2 , 3) (3 , 2) (4 , 3)
$\langle text\ analysis \rangle$	(2 , 4) (3 , 3) (4 , 4)
$\langle mining\ for \rangle$	(1 , 4) (2 , 3) (4 , 3)
$\langle mining\ analysis \rangle$	(2 , 4) (4 , 4)
$\langle for\ analysis \rangle$	(2 , 4) (3 , 3) (4 , 4)

Step 3: Generation of the frequent sequences of length greater than 2 We use the frequent 2-sequences to generate frequent sequences of length greater than two. The matching is based on the sequence-extension principle as shown in Algorithm 2.

For instance, the frequent 3-sequence $\langle text\ mining\ for \rangle$ can be generated by extending $\langle text\ mining \rangle$ with the 2-sequence $\langle mining\ for \rangle$ (*cf.* Table 4).

Thus, longer frequent sequences are iteratively derived starting from the frequent 2-sequences. For our running example, only one 4-sequence is found, which is $\langle text\ mining\ for\ analysis \rangle$.

Step 4: Pruning step We only retain *frequent closed sequences* since we look for compact set of term sequences by pruning redundant ones. Our pruning procedure is based on the backward super-sequence condition (*cf.* Proposition 1) which is tested on each candidate k-frequent sequence. Hence, if a sequence $S_{(k+n)}$ is a backward super-sequence of another sequence S_k, *i.e.,* they have the same support, then this latter is pruned from the set $\mathcal{FCS}$. For example, the frequent 3-sequence $\langle mining\ for\ analysis \rangle$ is a backward super-sequence of the 2-sequence $\langle for\ analysis \rangle$. Therefore, the sequence $\langle for\ analysis \rangle$ is discarded from the set $\mathcal{FCS}$.

4.2 *Inter-lingual association rules based on frequent closed sequences*

In what follows, we describe the way to derive from $\mathcal{FCS}$ the inter-lingual association rules, named ILAR-n-to-m, for phrase-based machine translation.

While considering frequent closed sequences sets $\mathcal{S}^{fr}$ and $\mathcal{S}^{en}$, an ILAR is an implication of the form: $R : S_{fr} \Rightarrow S_{en}$ such that S_{fr} and S_{en} are two frequent closed sequences of terms of lengths n and m, respectively. In machine

translation, this means that the English sequence S_{en} is a potential translation of the French sequence S_{fr}. Note that, we keep the same definitions for the support and the confidence of an ILAR as given in Equation (6).

In order to use inter-lingual association rules in statistical machine translation, we need to assign to each rule $R : S_{fr} \Rightarrow S_{en}$ a probability computed as follows:

$$\forall\, S_f \in \mathcal{S}^{fr},\ S_{e_j} \in \mathcal{S}^{en}, P(S_{e_j}|S_f) = \frac{Conf(S_f \Rightarrow S_{e_j})}{\sum\limits_{i\in[1\ldots n]} Conf(S_f \Longrightarrow S_{e_i})} \qquad (7)$$

We present in the next section the experimental evaluation of the two approaches described above in the context of phrase-based machine translation.

5 EXPERIMENTAL STUDY

All experiments are carried out on a part of the proceedings of the European Parliament EUROPARL [16]. The proposed models have been tested in a whole translation decoding system by using PHARAO decoder [4] and then compared to the performance of state-of-the-art both for word and phrase-based machine translation [17]. We use the BLEU (BiLingual Evaluation Understudy) score [18] for evaluation.

5.1 *Material*

We used a French-English parallel corpus of 596831 sentence pairs. Table 5 gives more details about the used parallel corpus EUROPARL.

Table 5. Quantitative description of the used corpus.

		French	**English**
Train	Sentences	$596K$	
	Words	$17.3M$	$15.8M$
	Singletons	$26.6K$	$22.2K$
	Vocabulary	$77.5K$	$60.3K$
Development	Sentences	1444	
	Words	$15.0K$	$14.0K$
Test	Sentences	500	
	Words	$5.2K$	$4.9K$

EXPERIMENTAL RESULTS The direction of translation is from English to French. Tests have been achieved on a corpus of 500 sentences. The phrase translation table of the state-of-the-art system, denoted in the remainder by Koehn-Och, is acquired from a word-aligned parallel corpus by extracting all phrase-pairs that are consistent with the word alignment [17].

Our method based on inter-lingual triggers retrieves from the source language a set of 11 212 pertinent phrases which are composed of two or three words, only 8.31% of these phrases occur in the test corpus. This percentage is very low in order to hope to noticeably improve the results.

Table 6 illustrates performances of different systems on both development and test corpora. On the development corpus, the use of pertinent $n.m$-Triggers improved the results achieved by 1.1-Triggers by 13.27%. For the state-of-the-art methods, the use of phrases increases the performance by 19.90% compared to the word-based method. On the test corpus, both methods improve the results by respectively 5% and 25%. This difference may be explained by the fact that the state-of-art method uses a translation table of more than 21 millions of entries (of one or more words) whereas ours uses only 5.2 millions (where the number of phrases do not exceed 20 000 phrases). Consequently, our translation table has a weak coverage of the training corpus. We should thus increase the size of the translation table in order to get closer values to those of the state-of-art one. By adding 1.71 millions of phrases, we achieve a BLEU result of 34.41. This improvement reduces the gap between our results and the state-of-art one to 2.74, knowing that our translation table is very small in comparison to the one used by Koehn-Och.

Table 6. Experimental evaluation in terms of the BLEU score on the development and the test corpora.

	Inter-lingual triggers		ILAR		State of the art	
	1.1	n.m	1-to-1	n-to-m	IBM model 3	Koehn-Och
Development	31.02	35.27	19.71	32.66	29.23	35.07
Test	30.97	32.75	22.06	34.18	29.57	37.15

As illustrated in Table 6, the phrase-based MT system based on ILARs fulfilled a BLEU score of 34.18. So, by adding pertinent frequent closed sequences, we achieved an improvement of more than 12 points in terms of BLEU compared to the initial word-based MT system which only considers ILAR between single terms (22.06). Note that, we do not restrict the length of the generated frequent closed sequences, although derived sequences from EUROPARL may reach a size of 25 terms. We also experimentally observed that beyond a certain length (sequences of 14 terms), the BLEU score does not increase. This could be explained by the fact that these sequences are very rare in the training corpus, *i.e.*, they have

a very low support, and their frequency in the test corpus is even lower. Therefore, they can not improve the BLEU score.

6 TABLE TRANSLATION MIXTURE

In order to take advantage of the two original methods presented above, we decided to combine their two translation tables. Because the $n.m$-Triggers table does not contain a great number of phrases and since 1.1-Triggers achieved better results than IBM 3 model, we decided to put in a new translation table the word translations got from 1.1-Triggers and phrases obtained by association rules, *i.e.*, ILAR-n-to-m. The result is presented in Table 7. The combination (the line referenced by *Comb*) outperforms both proposed methods. This result shows that we come closed to the result of Koehn-Och. We are just 1.63 below of the standard method. This illustrates that it is possible to retrieve pertinent phrases and their corresponding translations without any need of alignment which constitutes the advantage of the two original methods presented in this paper.

Table 7. Evaluation in terms of BLEU score on test corpus.

Model	**BLEU**
Koehn-Och	37.15
ILAR-n-to-m	34.18
$n.m$Triggers	34.41
$Comb$	35.52

Moreover, Figure 1 illustrates the evolution of the number of phrases in accordance to the number of words per phrase. We can see that the Koehn-Och curve has a cubic form [3] whereas ours has a decreasing power aspect [4]. The Koehn-Och curve grows until 5 and decreases for longer phrases. Whereas in our method the curve sharply decreases with the length of phrases.

Consequently, the gap between our result and the Koehn-Och one could be explained by the fact that the influence of phrases of four and five words would have a real impact. While the number of sequences of 5 words in Koehn-Och method is around 2.8 millions, in our table we got just 10500. Globally, our method produces pertinent phrases, however this number is not sufficient to reach the state-of-art result.

[3] Koehn-Och method: $y = -28953X^3 + 148491X^2 + 720014X - 90643$
[4] Our method: $y = 1.36E6X^{-3.6}$

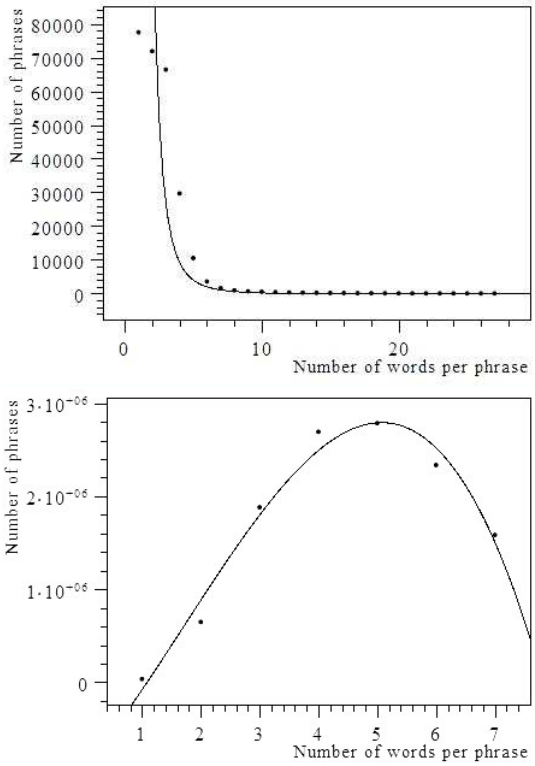


Fig. 1. Evolution of the number of phrases in *Comb* table (top) and Och-Koehn table (bottom).

7 CONCLUSION

In this paper, we proposed two new phrase-based machine translation methods. Each of them proposes a complete translation table containing translations of single words and phrases. The advantage of these methods is their easiness to develop statistical machine translation and more important than that, the fact that our methods, in contrast to what is done by the community, do not need any alignment.

We experimented our approaches on a large part of EUROPARL English-French language pair with a vocabulary of more than 60 000 linguistic units, and we evaluated them by using BLEU measure. Obviously, our methods have been compared to the pioneer ones. The results presented here are very encouraging.

They show that it is possible to consider the issue of statistical machine translation differently with the aim to improve the literature results. The advantage of our methods is that the selected phrases are pertinent but their number is not huge. In the future, we plan to improve our methods by making them selecting more sequences without losing the quality of translations.

REFERENCES

1. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Computational Linguistics **19**(2) (1993) 263–311
2. Och, F.J., Ney, H.: Improved statistical alignment models. In: Association of Computational Linguistics, Hongkong, China (October 2000) 440–447
3. Rosenfeld, R.: The CMU statistical language modeling toolkit and its use in the 1994 arpa csr evaluation. In: Proceeding of the Spoken Language Systems Technology Workshop, Austin (1995) 47–50
4. Koehn, P.: PHARAOH: a beam search decoder for phrase-based statistical machine translation models. In: Proceedings of Meeting of the American Association for Machine Translation (AMTA). (2004) 115–124
5. Koehn, P., al.: MOSES: Open source toolkit for statistical machine translation. Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstation session (2007)
6. Kohen, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedingd of the Human Language Technology and North American Association for Computaional Linguistics Conference, Edmonton (May-June 2003) 48–54
7. Rosenfeld, R.: Adaptive statistical language modeling: a maximum entropy approach. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh (1994)
8. Tillmann, C., Ney, H. In: Selection criteria for word trigger pairs in language modeling. Volume 1147. LNAI, Springer Verlag (1996) 98–106
9. Lavecchia, C., Smaili, K., Langlois, D., Haton, J.P.: Using inter-lingual triggers for machine translation. In: Proceedings of the eighth Conference Interspeech 2007, Antwerp, Belgium. (August 2007)
10. Latiri, C., Smaili, K., Lavecchia, C., Langlois, D.: Mining monolingual and bilingual corpora. Intelligent Data Analysis (IDA) **14**(6) (2010)
11. Och, F.J.: An efficient method for determining bilingual word classes. In: Proceedings of EACL, Bergen. (1999) 71–76
12. Zitouni, I., Smaili, K., Haton, J.P.: Statistical language modeling based on variable length sequences. Computer Speech and Language **17**(4-5) (2003) 27–41
13. Smaili, K., Brun, A., Zitouni, I., Haton, J.: Automatic and manual clustering for large vocabulary speech recognition : A comparative study. In: Proceedings of the European Conference on Speech Communication and Technology (EuroSpeech'99). (1999)

14. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD Conference, Washington, DC, USA. (May 1993) 207–216
15. Chang, K.Y.: Efficient sequential pattern mining by breadth-first approach. Master thesis, Available at http://web.management.ntu.edu.tw/chinese/im/theses/r92/r91725010.pdf, Accessed on september 21, 2010., National Taiwan University (2004)
16. Koehn, P.: EUROPARL: A multilingual corpus for evaluation of machine translation. In: Proceeding on the MT Summit, Thailand. (2005)
17. Och, F., Ney, H.: Discriminative training and maximum entropy models for statistical machine translation. In: 'Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA (July 2002) 295–302
18. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02). (2001) 311–318

CHIRAZ LATIRI
URPAH TEAM, COMPUTER SCIENCES DEPARTMENT,
FACULTY OF SCIENCES OF TUNIS, EL MANAR UNIVERSITY,
TUNISIA
E-MAIL: <CHIRAZ.LATIRI@GNET.TN>

KAMEL SMAÏLI
LORIA, SPEECH GROUP, VANDOEUVRE NANCY,
FRANCE
E-MAIL: <KAMEL.SMAILI@LORIA.FR>

CAROLINE LAVECCHIA
LORIA, SPEECH GROUP, VANDOEUVRE NANCY,
FRANCE
E-MAIL: <CAROLINE.LAVECCHIA@LORIA.FR>

CYRINE NASRI
URPAH TEAM, COMPUTER SCIENCES DEPARTMENT,
FACULTY OF SCIENCES OF TUNIS, EL MANAR UNIVERSITY,
TUNISIA
E-MAIL: <CYRINE.NASRI@GMAIL.COM>

DAVID LANGLOIS
LORIA, SPEECH GROUP, VANDOEUVRE NANCY,
FRANCE
E-MAIL: <DAVID.LANGLOIS@LORIA.FR>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 209–223
RECEIVED 01/11/10 ACCEPTED 21/12/10 FINAL 15/02/11

Meta-Evaluation of Comparability Metrics Using Parallel Corpora

BOGDAN BABYCH, ANTHONY HARTLEY

University of Leeds, UK

ABSTRACT

Metrics for measuring the comparability of corpora or texts need to be developed and evaluated systematically. Applications based on a corpus, such as training Statistical MT systems in specialised narrow domains, require finding a reasonable balance between the size of the corpus and its consistency, with controlled and benchmarked levels of comparability for any newly added sections. In this article we propose a method that can meta-evaluate comparability metrics by calculating monolingual comparability scores separately on the "source" and "target" sides of parallel corpora. The range of scores on the source side is then correlated (using Pearson's r coefficient) with the range of "target" scores; the higher the correlation – the more reliable is the metric. The intuition is that a good metric should yield the same distance between different domains in different languages. Our method gives consistent results for the same metrics on different data sets, which indicates that it is reliable and can be used for metric comparison or for optimising settings of parametrised metrics.

KEY WORDS: *Comparable corpora, machine translation, comparability metric, evaluation, subject domain, text genre.*

1 INTRODUCTION

In several areas of computational linguistics there is a growing interest in measuring the degree of 'similarity', or 'comparability', between

different corpora or between individual texts within these corpora. Interpretation of the concept of comparability varies according to the intended application, but many areas share the idea that it is useful to have an automated metric which ranks corpora, sub-corpora or documents according to the degree of their 'closeness' to each other. Typically, closeness is either measured by pre-defined formal parameters (such as lexical overlap) or intuitively described in terms of less formal linguistic categories (such as *genre* or *subject domain*). In the present study, formal metrics are based on combinations of measurable parameters that correlate with human intuitions about the intended linguistic categories.

The concept of comparability is relevant both in the monolingual context (as similarity between corpora/texts written in the same language) and in the cross-lingual context (as similarity of corpora/texts in different languages). Later we give examples of the areas and applications where measuring corpus and text comparability is useful.

In the monolingual context the concept of corpus comparability is used in computational lexicography for building translation dictionaries (e.g., Teubert, 1996 [1]), and in corpus linguistics for identifying qualitative differences between language varieties (e.g., British vs. American English), domains, modalities (spoken vs. written language), in order, for example, to determine which words are particularly characteristic of a corpus or text' (Kilgarriff, 2001:233 [2]; Rayson & Garside, 2000 [3]). Another monolingual application is automatic identification of domains and genres for texts on the web (e.g., Kessler et al., 1998 [4]; Sharoff, 2007 [5]; Vidulin et al., 2007 [6]; Kanaris & Stamatatos, 2009 [7]; Wu et al., 2010 [8]), with the goal of developing domain-sensitive and genre-enabled Information Retrieval (IR) methods, which can restrict search according to automatically identified fine-grained text types (such as blogs, forum discussions, editorials, analytical articles, news, user manuals, etc.).

Cross-lingual comparable corpora are frequently used for identifying potential translation equivalents for words, phrases or terminological expressions (Rapp, 1995 [9]; Rapp 1999 [10]; Fung, 1998 [11]; Fung & Yee, 1998 [12]; Daille & Morin, 2005 [13]; Morin et al., 2007 [14]), or supporting human translators in dealing with non-trivial translation problems (Sharoff et al., 2006 [15]; Babych et al., 2007 [16]). Multilingual comparable corpora are now becoming increasingly useful in training translation models for Statistical Machine Translation (SMT): (Wu & Fung, 2005 [17]; Munteanu et al, 2004 [18]; Munteanu & Marcu, 2006 [19]), especially for under-resourced languages, where

traditional parallel resources are not available, or are very small or in any other way unrepresentative (Vasiljevs, 2010 [20]). There are several dimensions of comparability, which can be summarised as follows:

(1) *granularity of comparability*: a measure of correspondence between units at different structural levels:

- corpus-level comparability – between corpus A and corpus B as a whole, or comparability between individual sections (subcorpora) within the corpus;
- document-level comparability – between different documents within or across corpora, e.g., (Lee et al, 2005 [21])
- paragraph- and sentence-level comparability – between structural and communicative units within or across individual documents, e.g., (Li et al., 2006 [22])
- comparability of sub-sentential units – between clauses, phrases, multiword expressions, lexico-grammatical constructions.

(2) *degrees of comparability*: a level of closeness between two units of comparison on the scale ranging between close, then free translations (or plagiarised sections in the monolingual context), then texts about the same event, texts on the same topic, corpora in the same domain, and finally to completely unrelated corpora. In the cross-lingual context we can distinguish the following broad categories:

- parallel corpora (consist of translated documents, where alignment at the sentence level is possible, e.g., corpora collected from multilingual news websites)
- strongly comparable corpora (consist of texts describing the same event or subject, where alignment at the document level is still possible, e.g., linked Wikipedia articles in different languages, news stories on the same event)
- weakly comparable corpora (consist of texts in the same domain or genre, but describing different events or areas; document alignment is usually not possible, e.g., collection of British and German laws on immigration policy).

Specific applications of comparable corpora use a different understanding and different ways of identifying the intended degree of closeness between corpora or texts. However, in many cases these metrics use similar sets of features and similar methods of calculating the scores. For example, both monolingual and cross-lingual

comparability in terms of subject domains typically rely on lexical features weighted or filtered by frequency, textual salience of key terms, etc. Often the only difference is that in the case of cross-lingual metrics lexical features (words) are mapped to words in another language using bilingual dictionaries or Machine Translation (MT) systems, while in monolingual applications lexical features are matched directly. These similarities mean that measuring comparability increasingly becomes a core technological challenge in Natural Language Processing that needs to be developed and evaluated systematically.

Many applications now require not just *it-looks-good-to-me* comparable corpora, but corpora with controlled and benchmarked levels of comparability according to certain criteria. Comparability metrics are used not only for reporting scores of closeness between corpora, but also for collecting additional texts to make a corpus bigger, or to filter out unwanted texts from corpora to ensure the intended level of comparability. From this perspective it is important to understand how reliable a particular metric is and to what extent it matches its specifications in its ability to *evaluate* comparability of corpora or individual texts. To date, we are not aware of any systematic research on such *meta-evaluation* (or calibration) of comparability metrics.

In this paper we give an example of an application domain which potentially requires corpora with controlled levels of comparability. We propose a method that can meta-evaluate different metrics used to measure comparability. We show that our method gives consistent results for the same metrics on different data sets, which indicates that it is reliable and can be used for selecting a best-performing metric, or for finding the most efficient parameters for parametrised metrics.

The rest of the paper is organised as follows. In Section 2 we describe our application area: creating a coherent selection of parallel texts for training domain-specific MT systems. In Section 3 (Methodology) we describe different metrics that we use for measuring corpus-level comparability and our methodology for meta-evaluation of these metrics. Section 4 presents the results of this meta-evaluation, and Section 5 discusses conclusions and future work.

2 APPLICATION OF CORPUS COMPARABILITY: SELECTING COHERENT PARALLEL CORPORA FOR DOMAIN-SPECIFIC MT TRAINING

Traditionally statistical and example-based MT have relied on parallel corpora (collections of texts translated by human translators) to train

statistical translation models and automatically extract equivalents. However, a serious limitation of this approach is that translation quality is impaired where parallel resources are not available in sufficient volume.

Firstly, it has been shown that on average improving translation quality at a constant rate requires an exponential increase in the training data (e.g., Och and Ney, 2003: 43) [23], i.e., if improving some MT evaluation score, e.g., BLEU, by one point required doubling the size of a training corpus, then further improvement by one additional point would require a corpus four times bigger than the initial size, etc. This dependency imposes fundamental limitations on translation quality even for well-resourced languages, such as English, German or French, where only the huge data sets used by engines like Google Translate produce relatively good quality (and even then, only for certain text types). Smaller and less resourced languages do not have the benefit of such data repositories, which results in a much lower MT quality.

Secondly, training translation models and language models for SMT has been shown to be domain-dependent to a much greater degree than rule-based MT (RBMT) (Eisele, 2008: 181) [24]. If an SMT engine is trained on a corpus that doesn't match the domain of the translated text, then the quality for such out-of-domain translation becomes much lower. In practice this means that for more narrow subject domains and text types SMT cannot produce acceptable translation quality without domain adaptation, which needs correspondingly highly-specific parallel textual resources.

For translation to and from under-resourced languages in narrow domains and for specific text types the two problems described above are combined. As a result traditional ways of building SMT engines with acceptable translation quality are often not possible for many domain / language combinations.

There is, therefore, a need to develop a fine-grained monolingual domain selection and domain control mechanism for evaluating comparability of corpus sections that can usefully be added to any SMT training corpus (comparability here is measured monolingually – either on the source or on the target side). The methodology should allow MT developers to balance the size of the corpus to be built and its internal consistency, in terms of how newly added sections match its originally intended subject domain.

3 METHODOLOGY

The methodology for computing the comparability of sections for an MT training corpus is typically based on calculating the degree of overlap between the two files in terms of simple word tokens, or at more advanced levels of linguistic annotations, such as lemmas (dictionary forms of words), combination of lemmas and part-of-speech codes, translation probabilities for each of the words, etc. There are several major challenges for the efficient calculations of this overlap.

Firstly, calculated scores for comparability should be consistent with human intuition about closeness between the two sections, and what constitutes the subject domain at different levels of granularity, e.g., the broader domain of computer hardware vs. a more narrow domain of network technologies, documentation for different types of network cards, etc. This is required if user needs for finer- or coarser-grained domains are to be adequately addressed for most types of projects.

Secondly, for practical applications the number of calculations between compared sections can be very large; so the calculation method should be fast enough to produce the results in real time.

Thirdly, comparison often needs to be done between sections of corpora file of different sizes, so the calculation method should be minimally affected by the size of the compared sections or texts.

Ideally, comparison should also take into account both source and target parts of new additions to corpora, and evaluate not only monolingual distance, but also the "translation" distance (which could mean that the same translation equivalents are used, and that terminology is translated consistently across the selected uploads).

3.1. DESCRIPTION OF CALCULATION METHOD

The method which we use in our experiments is based on the work of Kilgarriff (2001) [2]. This method was initially developed for the purposes of linguistic analysis, i.e., to find words which that are substantially different in two corpora, e.g., a corpus of spoken vs. a corpus of written English. But one of the side-effects of this method is that it can produce a single numeric value that shows the 'distance' between the two compared monolingual corpora.

So in our work we focus not on identifying individual words which are used differently in different corpora, but on general quantitative

measures of comparability between them. The method can be summarised as follows.

For each corpus (on the source or target side) we build a frequency list, and take the top 500 most frequent words (which also include function words).

Since corpora can be of different sizes, we use relative frequency (the absolute frequency, i.e., the number of times each word is found, divided by the total number of words in the corpus).

We compare corpora pairwise using a standard Chi-Square distance measure:

$$Chi^2 = \sum_{w_1}^{w_{500}} \frac{Freq(corpus^A) - Freq(corpus^B)}{Freq(corpus^A)}.$$

3.2. SYMMETRIC VS. ASYMMETRIC CALCULATION OF DISTANCE

The challenge for this method is that some words which are in the top-500 list for CorpusA may be missing from top-500 in CorpusB and vice versa. If the algorithm encounters the missing word, then it just adds its relative frequency to the value of the Chi-Square distance.

Obviously, exactly the same number of words is missing from the top-500 in CorpusA and in CorpusB. However, the sum of relative frequencies for these words can be different, e.g., it is possible that on average more frequent words will be missing from CorpusA, and less frequent from CorpusB.

Therefore, if we compute the Chi-Square distance from CorpusA to CorpusB, and then from CorpusB to CorpusA, we will get different values, which shows that the term 'distance' (used in an everyday sense) is not exactly right for describing the values: our calculation method is asymmetric.

Instead, Kilgarriff (2001) [2] uses a symmetric calculation: he takes into account only words which are common in both corpora, and goes down the frequency lists as far as it is needed to collect the 500 most frequent common words. This method always returns the same values of distance for any direction. However, the symmetric approach has its drawbacks: missing words do not contribute to the score directly (only by virtue of occupying 'someone else's place'); also it is harder to select the initial list for comparison: in the worst case scenario it is necessary to start with top 1000 words for each of the corpora, and then to remove mismatches. It may take slightly longer to do these calculations, and in

real time this may result in unnecessary delays and increased waiting time for users.

In our approach we make two independent asymmetric calculations in both directions: CorpusA → CorpusB and CorpusB → CorpusA, and get two Chi-Square scores.

However, now is not obvious what is the best way to combine these two scores into a single measure of distances between the corpora: one method would be to take the Average of the distances; another method is to take the Minimum distance as the value.

We experimentally compare these two possibilities in later sections, and show that the minimum of the two Chi-Square scores computed for the two directions gives better and more meaningful results.

3.3. CALIBRATING THE DISTANCE METRIC

We used corpora available from TAUS (Translation Automation User Society) in its TDA (TAUS Data Association) repository. Corpora there were initially annotated by data providers in terms of 'subject domains', which are identified manually at the upload stage. The idea is that the metric should be able to simulate identification of these domains automatically.

We calculated the distance between different sections of TDA repository – individual uploads and collections of uploads grouped by the same data provider and domain. In order to tell whether the metric intuitively makes sense, we checked whether there is an agreement between the resulting values and the labels provided by the TDA members.

In our experiment we focussed on the English (US and UK) – French (France) language pair. We selected the set of uploads in a way which covered different combinations of domains and data providers: some corpora are labelled as belonging to different domain, but were produced by the same company. Some were produced by different companies but were labelled with the same domain tag.

These labels were used as a benchmark for judging the quality of the lexical comparability metric. We aimed at giving the smallest distance score to corpora within the same subject domain.

The results of measuring comparability between sections of the corpus given by different data providers are presented in Figure 1. Different shades of grey visualise different ranges of distances: the closer the distance, the darker the colour.

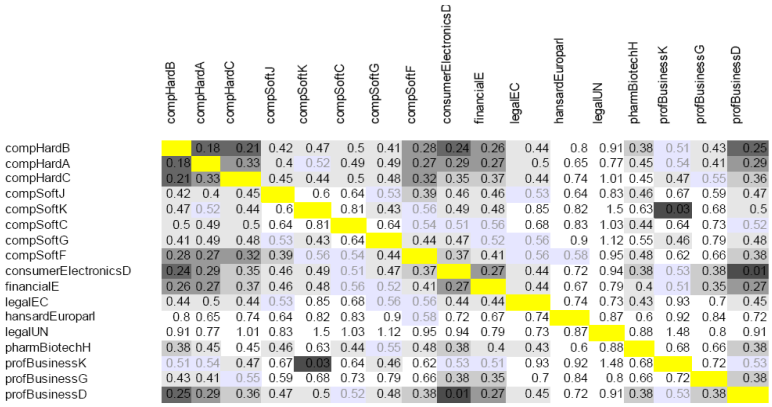


Fig. 1. Chi-Square distances between different data providers (labels indicate domain and owner, e.g., compSoftG is the label 'computer software' produced by the company G)

It can be seen from Figure 1 that the metric reliably identifies the following:

(1) all corpora within the 'computer hardware' domain are reliably grouped together, and distinguished from other domains;

(2) some of the corpora which were produced by the same companies 'D' and 'K' were reliably grouped together, even though the corpora had received different human labels: company D - 'consumer electronics' and 'professional business services'; company K - 'computer software' and 'professional business services'. These instances can be explained by inconsistency in assigning labels to corpora which essentially represented the same domain.

(3) different domains which are intuitively close to each other were also grouped together: 'computer hardware' and 'consumer electronics', and then at some distance – several corpora on computer software.

However, there are several problems with the presented distances and grouping:

(1) the 'computer software' and 'legal' domains are not coherently grouped. A possible reason is a greater variety of sub-domains within the 'computer software' domain (it may describe more 'products', and have more diverse lexical profiles);

(2) the ‘pharmaceutical and biotechnology' and ‘financial' domains are not sufficiently distinct from the ‘software' and ‘hardware' domains.

Still these problems can be attributed to inconsistencies in human

labelling, as well as to shortcomings of the metric itself. Symbolic labels are speculative in their nature, and do not capture the inner structure or diversity of the domain; at present human annotation offers no way of dealing with mislabelled data.

4. VALIDATION OF THE SCORES: CROSS-LANGUAGE AGREEMENT FOR SOURCE VS. TARGET SIDES OF TMX FILES

We validate our choice of metric by comparing different versions of Kilgarriff's metric for computing the distance between corpora. As we indicated, there are two possibilities for combining asymmetric Chi-Square distances: we can either take the average of the two different values, or go for the minimum of the two values.

Sections of corpora in the TDA repository are uploaded in TMX (Translation Memory Exchange) format, which is an XML file with sentence-or segment-aligned parallel corpora.

The idea for comparison is the following: we use each of the possibilities on the source side and on the target side of the same TMX files and then compare how the scores "agree" with each other. The agreement can be measured by standard statistical method for calculating correlation, like Pearson's r correlation coefficient: if there is a good agreement, r is closer to 1 or to -1; if there is no agreement r is closer to 0.

To get more data points for more reliable calculation of correlation we further split sections of the corpora presented in Figure 1 into individual uploads, e.g. for Hardware Company A we had 5 individual TMX files. Distances were computed at this finer granularity between all individual uploads.

If one of the compared metrics produces a higher correlation, then it means that the results obtained on the source side are more consistent with the results obtained on the target side, and the metric is more meaningful. Essentially, we know from the start that the two texts came from the same TMX, but the metric doesn't have that information. The better it can figure this out, the more reliable it is.

Table 1 compares the r correlation figures for individual uploads. It can be seeing from the table that the minimum distance has the best correlation between source and target sides of TMX: r=0.85, and can be viewed as a more reliable metric compared to the average distance or the third score we computed, the one-direction distance.

Table 1. Pearson's r correlation for distances computed for English vs French

Metric	r-correlation
Minimum distance	**0.85**
Average distance	0.67
One-direction distance (A → B and B → A)	0.61

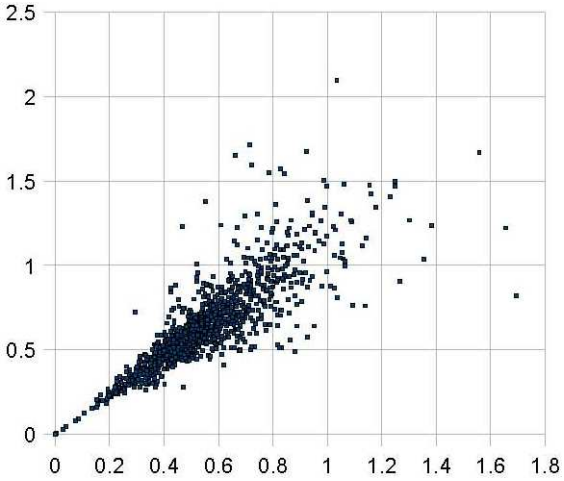


Fig. 2. Minimum Chi-Square distance (x = En; y = Fr)

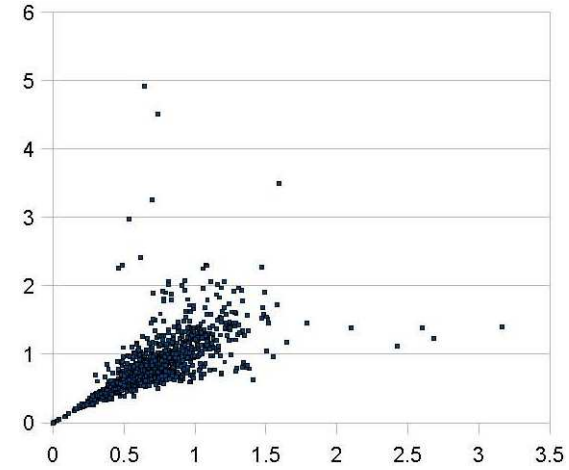


Fig. 3. Average Chi-Square distance (x = En; y = Fr)

Figures 2 and 3 further illustrate this difference: they compare the correspondence between the TMX distances for source text (horizontal axis) vs. distances for target texts for the same uploads (and illustrate the correlation figures presented above. Figure 2 indicates the distances in terms of minimum chi-square scores for TMX-A → TMX-B vs. TMX-B → TMX-A. Figure 3 indicates average chi-square scores for the same pairs of distances.

It can be seen that the minimum distances have a much better correlation between source and target, so they more reliably indicate whether the texts are indeed closer to each other.

This method offers a way to evaluate different comparability scores: the more the source and target agree with each other, the better the quality of the matching scores is. This evaluation method is based on the assumption that if the texts are close in terms of the source side, the scores should also show that they are close in terms of the target side.

However, there is a question of how to interpret divergences of the dots from the diagonal line. One explanation is that the quality of the matching scores is not so good. But another explanation suggests that some inconsistent translation equivalents are used across the upload in the target, so even if the documents are genuinely close on the source side, they become divergent on the target side. These issues require a more careful look into the compared data.

To verify that our meta-evaluation method provides consistent results for different sizes of evaluated corpus we repeated the experiment for the same language pair, but now we used the original joined TMX files, where all uploads are grouped together for the same data provider. Figure 3 shows an agreement for minimum chi-square scores for TMX-A → TMX-B vs. TMX-B → TMX-A. Correlation coefficients for this setting are presented in Table 2.

The highest correlation is again found between minimum scores across the two directions, which confirms our choice of this metric as the most reliable.

Table 2. Pearson's r correlation for English vs French on larger corpus sections

Metric	r-correlation
Minimum distance	**0.88**
One-direction distance (A → B)	0.72
One-direction distance (B → A)	0.86

These results shows that data with a different number of data points

obtained on sections of different sizes point to the same metric as the best one, which indicates that our proposed method is internally coherent.

5. DISCUSSION AND FUTURE WORK

We have proposed a method for meta-evaluation of comparability metrics using correlation between source and target sides of parallel corpora. We used a collection of parallel corpora available from the TDA repository. Comparability metrics need to be calibrated on a diverse parallel corpus that includes sections with several distinct and annotated subject domains, genres, etc. However, after successful calibration such comparability metrics can be further used to collect monolingual and bilingual comparable corpora, without the need to have expensive parallel resources. Pearson's r coefficient, which we calculate during calibration, gives an indication of how reliable the metric is and how much noise might occur in the data.

The metric which was found to perform best in our experiment (minimum Chi-Square distance between the compared top-500 words of frequency lists) has relatively high agreement for data generated on the source and target sides of TMX files (r=0.85), which indicates the upper limit of the metric's reliability.

However, applicability of the proposed meta-evaluation method is limited by the accuracy and completeness of translations in the parallel corpus used for calibration: gaps or excessively free translation can result in shifts in feature patterns, so distances between different domains calculated on source and target texts can become greater. Another potential limitation of the method is its reliance solely on those formal parameters which can be computed in a language-independent way, and do not vary substantially across languages. Language-specific differences, e.g., variations in type-token ratio (due to different morphological structure of languages) can potentially lead to differences in feature patterns and, as a result, lower correlation figures.

In future work we will investigate to what extent our method is limited by the quality of the translated parallel corpora and by language-specific features – by measuring the comparability of real corpora. We will also explore ways to externally validate the proposed method.

REFERENCES

1. Teubert, W. (1996). Comparable or Parallel Corpora? International Journal of Lexicography, 9, pp. 238-264.
2. Kilgarriff, A. (2001). Comparing Corpora. In: International Journal of Corpus Linguistics 6 (1): 1-37. (Reprinted in Corpus Linguistics: Critical Concepts in Linguistics. Teubert and Krishnamurthy, editors. Routledge. 2007), www.kilgarriff.co.uk/Publications/2001-K-CompCorpIJCL.pdf
3. Rayson, P. & Garside, R. (2000). Comparing corpora using frequency profiling. In: WCC '00: Proceedings of the workshop on Comparing corpora (pp. 1-6).
4. Kessler, B., Numberg, G. & Schuetze, H. (1998). Automatic detection of text genre. In: ACL'98: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (pp. 32-38).
5. Sharoff, S. (2007). Classifying Web Corpora into Domain and Genre Using Automatic Feature Identification. In: Proceedings of Web as Corpus Workshop.
6. Vidulin, V., Lustrek, M. & Gams, M. (2007). Using genres to improve search engines. In: Proceedings of the international workshop towards genre-enable search engines: The impact of natural language processing (pp. 45-51).
7. Kanaris, I. & Stamatatos, E. (2009). Learning to recognize webpage genres. Information Processing and Management, 45, pp. 499-512.
8. Wu, Z., Markert, K., Sharoff, S. (2010). Fine-Grained Genre Classification Using Structural Learning Algorithms. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (pp. 749-759).
9. Rapp, R. (1995). Identifying word translations in non-parallel texts. In: ACL '95: Proceedings of the 33rd annual meeting on Association for Computational Linguistics (pp. 320-322)
10. Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. In: ACL '99: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (pp. 519-526)
11. Fung, P. (1998). A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-parallel Corpora. In: Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA'98) (pp. 1-16). : Springer
12. Fung, P. & Yee, L.Y. (1998). An IR approach for translating new words from nonparallel, comparable texts. In: COLING '98: Proceedings of the 17th international conference on Computational linguistics (pp. 414-420)
13. Daille, B. & Morin, E. (2005). French-English Terminology Extraction from Comparable Corpora. In: IJCNLP (pp. 707-718).
14. Morin, E., Daille, B., Takeuchi, K. & Kageura, K. (2007). Bilingual Terminology Mining - Using Brain, not brawn comparable corpora. In

ACL-07 (pp. 664-671).
15. Sharoff, S., Babych, B. & Hartley, A. (2006). Using Comparable Corpora to Solve Problems Difficult for Human Translators. In: COLING/ACL 2006 Main Conference Poster Sessions (pp. 739-746).
16. Babych, B., Hartley, A., Sharoff, S. & Mudraya, O. (2007). Assisting Translators in Indirect Lexical Transfer. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (pp. 136-143)
17. Wu, D. & Fung, P. (2005). Inversion Transduction Grammar Constraints for Mining Parallel Sentences from Quasi-Comparable Corpora. Natural Language Processing IJCNLP 2005, 3651, pp. 257-268.
18. Munteanu, D. S., Fraser, A. & Marcu, D. (2004). Improved Machine Translation Performance via Parallel Sentence Extraction from Comparable Corpora. In: HLT-NAACL 2004: Main Proceedings (pp. 265-272)
19. Munteanu, D. S. & Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (pp. 81-88)
20. Vasiljevs, A. (2010) ACCURAT - using comparable corpora for MT. In: LT Days 2010, Luxembourg, 2010-03-22 – 2010-03-23, presentation available at: www.accurat-project.eu/uploads/publications/s7-vasiljevs.pdf
21. Lee, M. D., Pincombe, B. & Welsh, M. (2005). An empirical evaluation of models of text document similarity. In: Proceedings of the 27th Annual Conference of the Cognitive Science Society (pp. 1254-1259)
22. Li, Y., McLean, D., Bandar, Z., O'Shea, J. & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering, 18(8), pp. 1138-1150.
23. Och F. and Ney H. (2003) A systematic comparison of various statistical alignment models. In Computational Linguistics, 29(1):19-51, March.
24. Eisele, A., Federmann, C., Saint-Amand, H., Jellinghaus, M., Herrmann, T. & Chen, Y. (2008). Using Moses to Integrate Multiple Rule-Based Machine Translation Engines into a Hybrid System. In: Proceedings of the Third Workshop on Statistical Machine Translation (pp. 179-182).

BOGDAN BABYCH
CENTRE FOR TRANSLATION STUDIES,
UNIVERSITY OF LEEDS,
WOODHOUSE LANE, LEEDS LS16 5PT, UK
E-MAIL: <B.BABYCH@LEEDS.AC.UK>

ANTHONY HARTLEY
Centre for Translation Studies,
University of Leeds,
WOODHOUSE LANE, LEEDS LS16 5PT, UK
E-mail: <A.HARTLEY@LEEDS.AC.UK>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 225–237
RECEIVED 31/10/10 ACCEPTED 26/11/10 FINAL 11/02/11

Translating the Penn Treebank with an Interactive-Predictive MT System

MARTHA ALICIA ROCHA[1] AND JOAN ANDREU SÁNCHEZ[2]

[1] *Instituto Tecnológico de León, México*
[2] *Universidad Politécnica de Valencia, Spain*

ABSTRACT

The Penn Treebank corpus is a commonly used corpus in the Computational Linguistics community. This corpus is manually annotated with lexical and syntactical information. It has been extensively used for Language Modeling, Probabilistic Parsing, PoS Tagging, etc. In recent years, with the increasing use of Syntactic Machine Translation approaches, the Penn Treebank corpus has also been used for extracting monolingual linguistic information for further use in these Machine Translation systems. Therefore, the availability of this corpus adequately translated to other languages can be considered an challenging problem. The correct translation of the Penn Treebank corpus by using Machine Translation techniques and then amending the errors in a post-editing phase can require a large human effort. Since there is not parallel text for this dataset, the translation of this corpus can be considered as a translation problem in the absence of in-domain training data. Adaptation techniques have been previously considered in order to tackle this problem. In this work, we explore the translation of this corpus by using Interactive-Predictive Machine Translation techniques, that has proved to be very efficient in reducing the human effort that is needed to obtain the correct translation.

1 INTRODUCTION

The Penn Treebank corpus [1] is a very used corpus in many applications of Computational Linguistic. This corpus consists of approximately fifty

thousand sentences that were manually annotated with lexical information and syntactical information. This corpus has been used for many task related to Language Modeling [2–4], Probabilistic Parsing [5–8], PoS Tagging [9, 10], etc. One important advantage of this corpus is that its annotated information allows machine learning techniques to obtain very accurate and profitable linguistic information. Examples of these machine learning techniques are Maximum Entropy [10, 5], Probabilistic Estimation [4, 11] and Grammar Learning [2, 8]. Since this corpus is manually annotated and reviewed by human experts, it allows a reliable comparison with the gold annotation.

In the last years, syntax has become important for Machine Translation (MT) [12–15]. Some of these works are based on the availability of syntactically annotated corpus [13, 14, 16]. Most of these works use the Penn Treebank corpus for learning a syntactical model and then the Syntactic MT system is used in another task different from the Penn Treebank corpus. Therefore, a very interesting step forward in Syntactic MT would be to have the Penn Treebank correctly translated and apply Syntactic MT techniques to this translated corpus. In addition, another very challenging problem would be to use a parallel treebank to study new approaches of Syntactic MT.

The manual translation of the Penn Treebank corpus can be a very tedious and expensive task, and therefore it seems appropriate to carry out this task by using a MT system. The MT system can provide initial translations of the sentences, and a human could then review the full translation. This conventional post-editing review process for obtaining a high quality translation can be also expensive and tedious. In addition, it should be taken into account that there is not parallel text for the Penn Treebank corpus and the translation systems should be trained with out-of-domain data. Therefore, the translation quality of the output of the MT system could be low. Adaptation techniques for translating the Penn Treebank corpus were considered in [17] to alleviate this problem, but the final obtained results showed that a lot of errors were yet present in the final translations, and a lot of effort should be carried out to obtain correct high-quality translated sentences.

Therefore in this work we studied Interactive-Predictive MT (IPMT) techniques [18] for translating the Penn Treebank corpus. In IPMT, the human translator and the MT system work together in order to obtain correct high-quality translations. The human translator provides feedback to the MT translation and the system takes into account this feedback in order to constrain the search space and to avoid further errors. IPMT was

proven to be very effective for MT translation in [19] for in-domain tasks. But some tasks require to translate texts for which parallel text is not available. In such situation, the IPMT system starts with out-of-domain training data. However, [19] did not explore IPMT in the translation of an out-of-domain task. In this work we explored IPMT for translating the Penn Treebank corpus in which the initial system is trained with out-of-domain data.

2 INTERACTIVE-PREDICTIVE MACHINE TRANSLATION

Statistical MT has evolved rapidly in the last years, specially after the appearing the seminal papers of [20, 21]. Those MT systems were mainly based on words as basic unit translation. Currently, the state-of-the-art statistical MT systems use phrases as basic translation units [22, 23], although in recent years the syntax-based approach has provided very promising results [15].

In statistical MT, the problem can be stated as:

$$\widehat{\mathbf{y}} = \arg\max_{\mathbf{y}} \Pr(\mathbf{y}|\mathbf{x}) = \arg\max_{\mathbf{y}} \Pr(\mathbf{x}|\mathbf{y}) \Pr(\mathbf{y}), \qquad (1)$$

where $\mathbf{x}$ is a given input source sentence, $\Pr(\mathbf{y})$ is the language model probability and $\Pr(\mathbf{x}|\mathbf{y})$ is the translation model probability. The maximization is carried out over all possible output target sentences according to the language model. Expression (1) has been also stated in a different way by considering a *log linear* model [22]. In (1), both statistical models $\Pr(\mathbf{y})$ and $\Pr(\mathbf{x}|\mathbf{y})$ are usually trained on a very large training corpus.

For tasks in which both the training data and the test data are in the same domain, statistical MT systems that are based on the previously mentioned approaches are able to provide good translations results if the two languages share common structure and enough training data is available . The output sentences provided by the systems allow the user to understand the source sentence. However, the output sentence usually contains a lot of errors. If error-free translated sentences are required, then a human translator should review and correct the errors in a post-editing process. Interactive-Predictive MT (IPMT) intends to reduce the human effort that is needed to carry out this correction process.

In IPMT, the system and the user participate in a tight way in order to obtain the correct translation of an input sentence. First, the system provides an initial translation of the input sentence. Then, the user amends

the first detected erroneous word. This action implicitly involves to accept as correct a given prefix of the output sentence. This loop continues until the translation that the user has in mind is obtained. Every time the user provides a correction, the system incorporate the correction introduced by the user to the translation system in order to constrain the search space and to avoid further errors [18]. Next time that the system provides a new translation, it takes into account the correct prefix. In this way expression (1) is modified as follows:

$$\widehat{\mathbf{y}}_s = \arg\max_{\mathbf{y}_s} \Pr(\mathbf{y}_s|\mathbf{x}, \mathbf{y}_p), \tag{2}$$

where $\mathbf{y}_p$ and $\mathbf{y}_s$ are, respectively, the prefix and the suffix of the target sentence, and $\mathbf{y}_p$ includes the feedback provided by the user. Note that if $\mathbf{y} = \mathbf{y}_p\mathbf{y}_s$, the expression (1) and expression (2) are similar. This time, the search is carried out over the set of suffixes $\mathbf{y}_s$ that complete $\mathbf{y}_p$. Clearly, the feedback information $\mathbf{y}_p$ provided by the user is the opportunity to get better $\mathbf{y}_s$. The search process in the IPMT approach is carried out over a word graph. This word graph is obtained automatically after the system proposes the first hypothesis.

Figure 1 illustrates an example of how the IPMT system interacts with the human in a editing activity. In each iteration Iter-i the system uses a validated prefix $\mathbf{y}_p$ that it completes with a suffix $\mathbf{y}_s$ to compose the best hypothesis $\widehat{\mathbf{y}}$ of the following iteration. In the following iteration the user validates implicitly a new prefix $\mathbf{y}_p$ by typing an incorrect word $\mathbf{w}$, and again the system suggests a suitable continuation $\mathbf{y}_s$ for the following iteration. This process is repeated until a complete translation of the source sentence is reached. In the final translation, the 3 words typed by the user are underlined. In this example the estimated post-editing effort would be 13/23 (57%), produced by the errors: insert *"de"*, remove *"juntará el tablero"*, insert *"se unirá a la junta"*, remove *"29"*, and insert *"el 29 de"*. The corresponding interactive estimate is 3/23 (13%). This results in an estimated user effort reduction of 77%.

In the example that we have showed above the user corrects an error every time by typing a new correct word. If the new composed prefix that includes the typed word is not in the word graph, then the most probable path is computed by using an error-correcting algorithm. In [19], other ways of amending errors were studied. They proposed to use Mouse Actions (MA) as an additional feedback in order to obtain the correct translation. These MA could be of two ways: non-explicit positioning MA and interaction-explicit MA. In *non-explicit positioning MA*, when the

Source (x) :		Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov. 29 .
Target (y):		Pierre Vinken , de 61 años de edad , se unirá a la junta como director no ejecutivo el 29 de noviembre .
Iter-0	$\mathbf{y}_p$	()
	$\mathbf{y}_s$	Pierre Vinken , 61 años de edad , juntará el tablero como director no ejecutivo noviembre 29 .
Iter-1	$\widehat{\mathbf{y}}$	Pierre Vinken , 61 años de edad , juntará el tablero como director no ejecutivo noviembre 29 .
	$\mathbf{y}_p$	Pierre Vinken ,
	$\mathbf{w}$	de
	$\mathbf{y}_s$	61 años de edad , juntará el tablero como director no ejecutivo noviembre 29 .
Iter-2	$\widehat{\mathbf{y}}$	Pierre Vinken , de 61 años de edad , juntará el tablero como director no ejecutivo noviembre 29 .
	$\mathbf{y}_p$	Pierre Vinken , de 61 años de edad ,
	$\mathbf{w}$	se
	$\mathbf{y}_s$	unirá a la junta como director no ejecutivo noviembre 29 .
Iter-3	$\widehat{\mathbf{y}}$	Pierre Vinken , de 61 años de edad , se unirá a la junta como director no ejecutivo noviembre 29 .
	$\mathbf{y}_p$	Pierre Vinken , de 61 años de edad , se unirá a la junta como director no ejecutivo
	$\mathbf{w}$	el
	$\mathbf{y}_s$	29 de noviembre .
Final	$\widehat{\mathbf{y}}$	Pierre Vinken , de 61 años de edad , se unirá a la junta como director no ejecutivo el 29 de noviembre .
	$\mathbf{w}$	#
	$\mathbf{y}_p \equiv \mathbf{y}$	Pierre Vinken , de 61 años de edad , se unirá a la junta como director no ejecutivo el 29 de noviembre .

Fig. 1. Simulated example of IPMT interaction to translate the sentence of the Penn Treebank *"Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov. 29 ."*.

user positions the cursor in the place where he wants to type a new word, he is implicitly indicating that the word after the cursor is incorrect. Then, the search engine can start to look for a new suffix in which the first word is different from the current word after the cursor. In *interaction-explicit MA*, the user asks for a new suffix each time he presses the mouse. Each new suffix has to start with a word different from initial words that have appeared in previous rejected suffixes. This is formalized as follows:

$$\widehat{\mathbf{y}}_s = \underset{\mathbf{y}_s:\mathbf{y}_{s_1} \neq \mathbf{y}_{s_1}^{(i)} \forall i \in \{1..n\}}{\arg\max} \Pr(\mathbf{y}_s|\mathbf{x}, \mathbf{y}_p, \mathbf{y}_s^{(1)}, \mathbf{y}_s^{(2)}, \ldots, \mathbf{y}_s^{(n)}), \quad (3)$$

where $\mathbf{y}_{s_1}^{(i)}$ is the first word of the *i*-th suffix discarded and $\mathbf{y}_s^{(1)}$, $\mathbf{y}_s^{(2)}$, ..., $\mathbf{y}_s^{(n)}$ are the n suffixes discarded. Note that in some sense the *non-explicit positioning MA* approach is included in the *interaction-explicit MA* approach.

In the following section we present experiments in which we use an IPMT framework for the translation of the Penn Treebank corpus.

3 EXPERIMENTS

In this section we describe the experiments that we carried out to translate the Penn Treebank using an IPMT system with the interaction-explicit MA approach previously described.

3.1 *Datasets*

The Penn Treebank corpus is an annotated corpus that has approximately $49,000$ sentences. From this corpus, we used $1,141$ sentences from section 23 for the experiments, since this section is usually used for testing. These sentences were manually translated to Spanish by human experts without the help of any MT system: 500 of them were obtained from [17], and the other 641 were obtained from [24][3]. The source sentences of both datasets did not overlap each other. The 500 translated sentences from [17] were reviewed by two native Spanish speakers. The 641 sentences from [24] were translated by other human experts different from [17]. We called the $1,141$ sentences dataset Small Parallel Penn Treebank set (SPPT), we called SPPT-R the dataset from [17], and we called SPPT-M

[3] This translated corpus is available at `http://www.dsic.upv.es/~jandreu/SPTB.tgz`

the dataset from [24]. Note that the SPPT dataset was the union of the SPPT-R and SPPT-M datasets. We distinguished between SPPT-M and SPPT-R because they were translated in different places and in different contexts, and we wanted to check if there were notably differences in the translation experiments due to these differences. Table 1 presents the main characteristics of these datasets.

Table 1. Characteristics of the SPPT-R, SPPT-M, and SPPT datasets.

	SPPT-R		SPPT-M		SPPT	
Total of	**English**	**Spanish**	**English**	**Spanish**	**English**	**Spanish**
Sentences	500	500	641	641	1,141	1,141
Running Words	12,172	13,175	15,133	16,847	27,305	30,022
Vocabulary	2,613	2,946	3,613	4,120	4,918	5,862

Since there is no in-domain parallel data to train a system for translating the Penn Treebank, we had to use an out-of-domain parallel text. We used the second version of the *Europarl* bilingual corpus [25]. This corpus was used for training a phrase-based translation model. For these experiments we used only the sentences that had less than or equal to 40 words. The main characteristics of this training set can be seen in Table 2. All the experiments were carried out with uncapitalized text and appropriately tokenized.

Table 2. Characteristics of *Europarl* corpus.

Total of	**English**	**Spanish**
Sentences	730,740	730,740
Running Words	15,242,854	15,702,800
Vocabulary	64,076	102,821

3.2 *Assessment Metrics*

For assessment, we used some of the metrics defined and used in [18, 19]. The quality of the interactive translation is given by the Word Stroke

Ratio (WSR) defined as the number word-strokes that a user would need to perform in order to obtain the reference translation divided by the total number of words in the reference sentence. Note that a word-stroke is considered as a single action. We expect that the number of word strokes that are necessary to obtain the correct translation decreases in the IPMT system as the prediction changes their predictions and provides more accurate suffixes. Note that the same metric can be used for a non IPMT, and in such case it is similar to the usual Word Error Rate (WER) metric. For this reason we used this metric also for the post-editing experiments.

Another metric that we used was the Word Click Ratio (WCR) defined as the number of mouse actions per word that the user had to perform before accepting a new prediction with respect to using exclusively the keyboard in IPMT system.

The also measured the Effort Reduced (ER) as the relative difference between two evaluations in WSR.

3.3 *Results*

For the IPMT experiments, first of all, an English-Spanish phrase-based MT system was built. This was carried out by means of the public software THOT[4], GIZA++[5], and SRILM[6]. THOT and GIZA++ were used for training the translation model ($\Pr(\mathbf{x}|\mathbf{y})$ in expression (1)), and SRILM was used for obtaining a 5-gram language model ($\Pr(\mathbf{y})$ in expression (1)). A multi-stack decoder [26] was used to generated the word graphs and the hypotheses. Note that no process was carried out in order to adjust the parameter of the log-lineal model, because in such case we should leave some data for development and the dataset was not very large.

Experiments were carried out with the three datasets previously described, that is, the SPPT, SPPT-R, and SPPT-M datasets. In all the experiments we computed the WSR, WCR and ER metrics. Table 3 shows the obtained results. We compared two scenarios: first scenario corresponds to a classical post-editing scenario (column Post-edit in Table 3). Second scenario corresponds to an IPMT scenario in which the user carries out just a single MA (column IPMT in Table 3). Column ER shows an estimation of the percentage of ER achieved by using the IPMT scenario with respect to post-editing scenario. It is important to note that the

[4] `http://sourceforge.net/projects/thot/`

[5] `http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html`

[6] `http://www-speech.sri.com/projects/srilm/`

percentage of ER increased for the three datasets. Note also that the results in the post-editing scenario reflect some lexical differences between SPPT-R and SPPT-M. Values in column Post-edit reflects that the initial hypothesis provided by the system had a lot of error, but it must be taken into account that the models were trained on out-of-domain data. *Europarl* is a dataset related to speech transcription of the European parliament. However, Section 23 of Penn Treebank contains a lot of stock market jargon and proper nouns, ant therefore a lot of out-of-vocabulary words.

Table 3. Obtained results (in percentage) with the SPPT, SPPT-R, and SPPT-M datasets using conventional post-editing against IPMT with a single MA.

	Post-edit	**IPMT**	
Dataset	**WER**	**WSR**	**ER**
SPPT-R	70.3	61.2	13.1
SPPT-M	74.3	65.5	11.8
SPPT	72.5	63.6	12.3

It is important to remark that in these experiments we obtained slightly better results than those reported in [19] for an analogous experiment. Thus, Figure 4 in [19], reported ER reduction with just one click of about 10% when translating from Spanish to English but starting from 60%, 7% from German to English starting from 70% of WER, and 10% when translating from French to English starting from 60% of WER. No experiment was reported in [19] from English to Spanish.

Figure 2 shows the WSR, WCR and ER for only the SPPT dataset as a function of the maximal number of MA allowed per incorrect word by the user before writing the correct word. We omitted the other datasets because the results were very similar. Point 0 in the WSR plot corresponds approximately to the conventional post-editing scenario, and coincides with row SPPT in column Post-edit of Table 3; point 1 coincides with IPMT column of the same row.

The difference between point 0 and point 1 in ER plot corresponds to the percentage 12.3 in Table 3. Note that the WSR decreased notably with just four MA. This reduction is along the line of the results reported in [19], or slightly better. WCR plot shows that for the number of average MA per word kept almost constant as the number of maximum MA

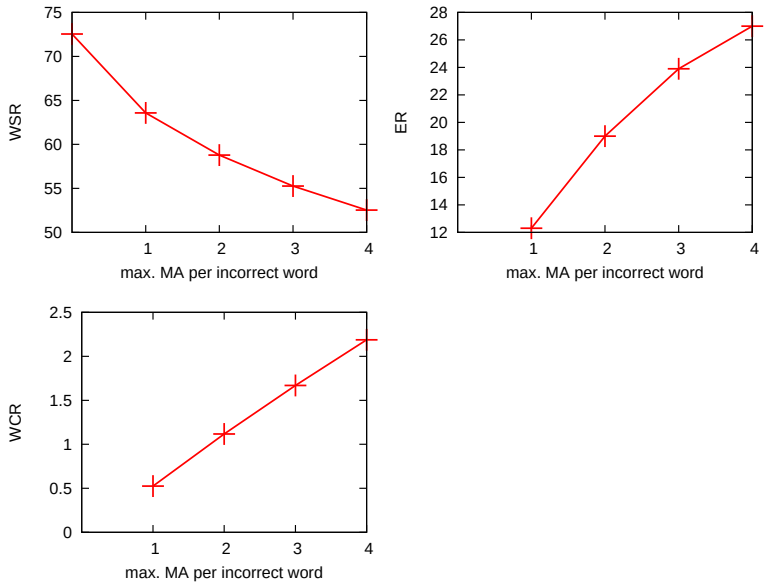


Fig. 2. WSR (top left), ER (top right) and WCR (bottom left) as a function of the maximal clicks of MA allowed by the user before write a new word.

allowed increased. Thus, for just 1 MA the quotient between WCR and max. MA was 0.52 (0.52/1.0), and for 4 MA this value was 0.54 (2.18/4).

A good trade-off is obtained when the maximum number of clicks is around 2 clicks, because a significant amount of effort is saved with a low number of extra clicks per word. These results showed that an adequate number of clicks to improve efficiency and reduce post-editing effort properly is a maximum of 2 or even 3 MA.

4 CONCLUSIONS

In this work we studied the use of IPMT for translating the Penn Treebank corpus. We followed the IPMT approach explored in [19] in which MA were considered as an input modality. We explored those ideas for a task in which there is not in-domain training data, and therefore out-of-domain training data had to be used. We proved that this input modality was informative enough in order to the obtain large human effort reduction even in this context. As a final comment, we can conclude that the

IPMT framework is a realistic approach to obtain high-quality translations in absence of in-domain training data.

Note that IPMT is stated for an scenario in which an expert translator collaborates on-line with a MT system in order to provide high quality translations. The user translates a sentence each time. Therefore, for future work we intend to explore learning techniques that make easier the use of the IPMT framework in order to deal with the translation of the Penn Treebank corpus. Thus, some techniques that we consider for future include Active Learning techniques [27, 28] that would allow us to select the sentences to be translated each time, and Online Learning techniques [11, 29] that would allow the models to be learned each time a sentence was translated.

ACKNOWLEDGEMENTS This work has been partially supported by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV program "Consolider Ingenio 2010" (CSD2007-00018), the MITTRAL project (TIN2009-14633-C03-01), and the Generalitat Valenciana under grant Prometeo/2009/014. The first author is supported by "División de Estudios de Posgrado e Investigación" and by "Metrología y Sistemas Inteligentes" research group of Instituto Tecnológico de León.

REFERENCES

1. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a large annotated corpus of english: the penn treebank. Computational Linguistics **19**(2) (1993) 313–330
2. Chelba, C., Jelinek, F.: Structured language modeling. Computer Speech & Language **14**(4) (2000) 283–332
3. Charniak, E., Knight, K., Yamada, K.: Syntax-based language models for statistical machine translation. In: Proc. of MT Summit IX, New Orleans, USA (September 2003)
4. Benedí, J., Sánchez, J.: Estimation of stochastic context-free grammars and their use as language models. Computer Speech & Language **19**(3) (2005) 249–274
5. Charniak, E.: A maximum-entropy-inspired parser. In: Proc. of NAACL-2000. (2000) 132–139
6. Roark, B.: Probabilistic top-down parsing and language modeling. Computational Linguistics **27**(2) (2001) 249–276
7. Collins, M.: Head-driven statistical models for natural language parsing. Computational Linguistics **29**(4) (2003) 589–637

8. Klein, D., Manning, C.D.: A* parsing: Fast exact viterbi parse selection. In: In Proceedings of HLT-NAACL 03. (2003) 119–126
9. Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. Comput. Linguist. **21**(4) (1995) 543–565
10. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Proc. Empiriral Methods in Natural Language Processing, University of Pennsylvania (May 1996) 133–142
11. Liang, P., Klein, D.: Online em for unsupervised models. In: Proceedings of HLT-NAACL 09, Stroudsburg, PA, USA (2009) 611–619
12. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Comput. Linguist. **23** (September 1997) 377–403
13. Yamada, K., Knight, K.: A syntax-based statistical translation model. In: In: Meeting on Association for Computational Linguistics, Stroudsburg, PA, USA (2001) 523–530
14. Yamada, K., Knight, K.: A decoder for syntax-based statistical mt. In: In: Meeting on Association for Computational Linguistics, Stroudsburg, PA, USA (2002) 303–310
15. Chiang, D.: Hierarchical phrase-based translation. Computational Linguistics **33**(2) (2007) 201–228
16. Petrov, S., Haghighi, A., Klein, D.: Coarse-to-fine syntactic machine translation using language projections. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, Association for Computational Linguistics (October 2008) 108–116
17. Rocha, M.A., Sánchez, J.A.: Machine translation of the penn treebank to spanish. In: I Joint SIG-IL/Microsoft Workshop on Speech and Language Technologies for Iberian Languages, Porto Salvo, Portugal (3-4 September 2009)
18. Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Ney, A.L.H., Tomás, J., Vidal, E.: Statistical approaches to computer-assisted translation. Computational Linguistics **35**(1) (2009) 2–28
19. Sanchis-Trilles, G., Ortiz-Martínez, D., Civera, J., Casacuberta, F., Vidal, E., Hoang, H.: Improving interactive machine translation via mouse actions. In: EMNLP 2008: conference on Empirical Methods in Natural Language Processing, Waikiki, Honolulu, Hawaii (October 25 – 27 2008)
20. Brown, P.F., Cocke, J., Della Pietra, S., Della Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics **16**(2) (1990) 79–85
21. Brown, P., Pietra, S.D., Pietra, V.D., Mercer, R.: The mathematics of statistical machine translation: parameter estimation. Computational Linguistics **19**(2) (1993) 263–311
22. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. Comput. Linguist. **30** (December 2004) 417–449

23. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proc. of the ACL Companion Volume Proceedings of the Demo and Poster Sessions. (June 2007) 177–180
24. Mellebeeck, B.: TransBooster: Black Box Optimisation of Machine Translation Systems. Phd thesis, School of Computing, Dublin City University (2007)
25. Koehn, P., Monz, C., eds.: Proceedings on the Workshop on Statistical Machine Translation. Association for Computational Linguistics, New York City (June 2006)
26. Ortiz, D., García-Varea, I., Casacuberta, F.: Generalized stack decoding algorithms for statistical machine translation. In: Proceedings on the Workshop on Statistical Machine Translation, HLT-NAACL 2006, New York City, USA (June 4–9 2006) 64–71
27. Burr, S., Mark, C.: An analysis of active learning strategies for sequence labeling tasks. In: Proc. of the Conference on Empirical Methods in Natural Language Processing EMNLP, ACL press (2008) 1069–1078
28. Haffari, G., Roy, M., Sarkar, A.: Active learning for statistical phrase-based machine translation. In: Human Language Technologies: The 2009 Annual Conference of the North America Chapter of the ACL, ACL press (June 2009) 415–423
29. Ortiz-Martinez, D., Garcia-Varea, I., Casacuberta, F.: Online learning for interactive statistical machine translation. In: Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT). (Jun 2010) 173–177

MARTHA ALICIA ROCHA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN,
INSTITUTO TECNOLÓGICO DE LEÓN,
MÉXICO
E-MAIL: <MROCHA@DSIC.UPV.ES>

JOAN ANDREU SÁNCHEZ
INSTITUTO TECNOLÓGICO DE INFORMÁTICA,
UNIVERSIDAD POLITÉCNICA DE VALENCIA,
SPAIN
E-MAIL: <JANDREU@DSIC.UPV.ES>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 239–255
RECEIVED 01/11/10 ACCEPTED 26/11/10 FINAL 08/02/11

Designing an Improved Discriminative Word Aligner

NADI TOMEH, ALEXANDRE ALLAUZEN,
THOMAS LAVERGNE, AND FRANÇOIS YVON

LIMSI/CNRS and Univ. Paris Sud, France

ABSTRACT

The quality of statistical machine translation systems depends on the quality of the word alignments, computed during the translation model training phase. IBM generative alignment models, despite their poor quality compared to a gold standard, perform well in practice. In this paper, we propose an improved word aligner based on a maximum entropy alignment combination model, which employ better feature engineering, ℓ^1 regularization, and an enhanced search space to improve the quality of both alignment and translation. For the Arabic-English language pair, we are able to reduce the Alignment Error Rate by 43.4%, and achieve ≈ 1 BLEU point enhancement over the IBM model 4 symmetrized alignments. These improvement are attainable at a lower computational cost, using only easy to estimate HMM and IBM model 1 features. An analysis of the obtained results shows that a good balance between several alignment characteristics should be maintained in order to deliver good translation quality.

1 INTRODUCTION

Word alignment aims to find word-level, non-compositional translational equivalences between two parallel sentences. In phrase-based, finding the optimal set of phrase pairs, which represents the translation model, is NP-hard [1]. To simplify the problem, constraints from a pre-computed word alignment are applied to restrict the search space, from which an extraction heuristic build the phrase-table. This two-steps approach makes the

problem of phrase pairs extraction boils down to the problem of word alignment, for which a wide range of models have been explored in the literature. Generative IBM models [2] are widely used in practice to construct two directional, one-to-many alignments in both translation directions, that are then symmetrized, on a sentence level, during a heuristical, post-processing step [3]. Training these models requires only a sentence-aligned bi-text, and is performed with the EM algorithm.

For linguistically different languages such as Arabic and English, a discriminative approach to word alignment is shown to be more effective even with a limited amount of labeled data. Indeed, the discriminative framework allows to model arbitrary and possibly inter-dependent aspects of the alignment process. In [4–6] word alignments is considered as a classification task, in which a binary classifier predicts for each possible assignment whether it should be included in the alignment or not. Discriminative models constitute a replacement to the local symmetrization heuristic that learns decisions in light of a global view of the data, by employing an arbitrary set of features including other models' predictions. Within this approach, the model extend the concept of symmetrization of two alignments into a combination of several ones.

However, in order to obtain a competitive performance, discriminative models face two issues that prevent their outspread application in practice. (1) The necessity to employ features based on predictions of IBM model 4 alignments, which are computationally demanding, and (2) technical issues (memory consumption and training/inference time) arising when incorporating a large number of features. In this paper, we extend the alignment combination and matrix modeling framework presented in [6] with an improved features engineering, combined with the use of ℓ^1 regularization for training the maximum entropy classifier. This kind of regularization allows the manipulation of a large number of features which will be selected during the training step. The resulting model is thus more compact and achieves similar results. Moreover, an improved search space is also investigated in order to increase the recall.

Using only easy to compute and exact models (IBM1 and HMM) as input, we are able to improve both alignment and translation quality, over the baselines. The improved search combined with stacking techniques yield the best performance. Three translation tasks of different sizes were considered to validate our findings. In order to shed some light on the nature of the relation between alignment and translation, we analyze BLEU scores in terms of alignment quality and other characteristics described in [7].

The rest of the paper is organized as follows. Related work is previewed in section 2 while the model with its components are presented in section 3. Finally, experiments and results are discussed in section 4.

2 Related Work

Several approaches for word alignment have been carried out recently and can be categorized in two major streams. In the first one word alignments is considered as a sequence labeling task, in which source words are tagged with target positions, using either generative models like HMM and IBM models [8] or discriminative ones like linear chain conditional random field (CRF) [9]. This representation of the problem results in directional, that require an additional symmetrization step to derive the many-to-many alignments. Symmetrization heuristics [8, 3], which starts with the intersection points of two directional alignments and progressively adds points from the union to cover unaligned words, performs well in practice. Even better performance can be achieved by tightly integrating symmetrization and model training [10].

The other stream aims to model the alignment matrix directly and produce many-to-many alignments, either employing generative models [11–14] or discriminative ones [15–18]. In the later, methods attempt to reach a good balance between the expressivity of the model and its complexity, in terms of tractability and the possibility of performing exact inference and learning. In [19], an alignment between two sentences is evaluated with a global score using a non-decomposable discriminative scoring function. This model resorts to a beam search since no restriction on the form of the resulting alignments is considered and the search space is intractable. In [20], tractability is achieved by casting the word alignment task as a maximum weighted matching problem, at the price of constraining possible alignments to one-to-one matchings and making local decisions with no global interactions. These limitations are fixed in [21], by modeling alignment as a quadratic assignment problem which is NP-hard in general. Word alignment is also casted as a structured classification problem, in which a decision must be made to activate (or disactivate) each cell of the alignment matrix, admitting some dependency structure between decisions. In [18] a CRF with a complex structure is used to predict the alignment, with approximate inference and a complicated two-step training. In [4] an independence assumption helps simplifying the problem while sacrificing the ability to model interactions between decisions. A middle ground solution is proposed in [6], where exact learning

and inference is insured within the maximum entropy framework, while interactions are modeled by an additional stacked classification layer. Our work falls in this framework with improved feature engineering, ℓ^1 regularization for ME training, and enhanced search space.

3 WORD ALIGNMENT AS A STRUCTURED PREDICTION PROBLEM

Following [6] we represent the task of word alignment as a structured classification one, where we aim to predict the alignment matrix using a maximum entropy classifier. We discuss the impact of the search space and regularization on obtained alignments.

3.1 *Maximum Entropy Classifier for Word Alignment*

Let $\mathbf{f}_1^I = f_1, f_2, \ldots, f_I$ and $\mathbf{e}_1^J = e_1, e_2, \ldots, e_J$ be a source and a target sentence, respectively. The task of word alignment is to find a a mapping between subsets of **f** and subsets of **e** (a many-to-many correspondence between words of **f** and words of **e**).

Alignment information between both sentences are represented by an alignment matrix $\mathbf{A} = \{l_{i,j} : 1 \leq i \leq I, 1 \leq j \leq J\}$, in which a particular link $l_{i,j}$ is considered to be *active* if the source word f_i is aligned to the target word e_j, and *inactive* otherwise. Thus, word alignment can be seen as a structured binary classification task. We employ a maximum entropy (ME) classifier to estimate the probability of a link of **A**:

$$p(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(y, \mathbf{x})\right),$$

where **x** denotes the observation, $Z(\mathbf{x})$ is a normalization constant, $(f_k)_{k=1}^K$ defines a set of feature functions, and $(\lambda_k)_{k=1}^K$ the associated set of weights.

In order to incorporate structure and dependencies into the ME model, without sacrificing efficiency, we use a *stacked generalization* method [22] which has been successfully applied to NLP problems [23], including word alignment [6].

In stacked learning, all labels are jointly predicted in two-steps, using two classifiers. The *second-level* classifier is trained using data *extended* with the predictions of the *first-level* classifier, which characterizes the dependency between labels. The task of word alignment implements this concept using a first pass aligner and uses its prediction as features to train the second pass aligner.

A *K*-fold selection process is employed to build training data for the global classifier [6].

During inference, the model assigns a probability to each proposed alignment link. The final output matrix consists of active links whose probability exceeds a threshold *p* (optimized on a development set using a grid search). This parameter is used to control the density of the resulting alignment and therefore the balance between its precision and recall. It also helps marginalizing the impact of the class-imbalance problem described below.

IMBALANCED DATASET Since the alignment matrix is typically sparse, with a majority of inactive links, the classification task we consider is imbalanced due to bias in training data acquisition. Whenever a class is over-represented its a priori probability to be chosen is higher than that of under-represented classes. Hence, attention should be payed to avoid learning a biased classifier with high tendency toward labeling all links as inactive. In previous work, the union of all input alignments is used to prune the search space and induce a more balanced dataset by reducing the number of links to be predicted to a subset of the alignment matrix: only points that have been proposed by at least one input alignment are labeled by the classifier, the others are assumed to be *inactive*. This reduction of the search space implies an upper bound on the recall by excluding a lot of plausible links, which become unreachable by the model. Since the ME classifier performs well precision-wise [6], the recall upper bound becomes a bottleneck.

ENHANCED ALIGNMENT SEARCH SPACE To push up the upper bound on recall, we exploit the observation that good candidate alignment points neighbor other good alignment points. Then the search space can be extended with additional links residing in a window of a fixed size, neighboring links proposed by input alignments. A down side for this heuristic is the increased number of negative examples, which shifts away training data from balance point. Possible solutions include random sub-sampling of the training data, and adjusting the selection threshold to neutralize the a priori probability assigned to over-represented *inactive* class.

TRAINING AND REGULARIZATION. The model is trained to optimize the regularized log-likelihood of the parameters. The most common regularization used in literature is the Gaussian prior (ℓ^2 penalty) which re-

duces overfitting and thus improve performance on most tasks. An alternative is to use a Laplacian prior (or ℓ^1 penalty). Such regularizer allows an efficient feature selection and yields sparse parameter vectors [24]. The regularization hyper-parameter aims to balance the pruning effect on the trained model.

To optimize the regularized log-likelihood, we use a second order quasi-Newton method. This kind of method requires a fully derivable function to optimize, which is not the case at zero for the ℓ^1 penalty. To overcome this problem, we use an adaptation of the classical L-BFGS, called OWL-QN, proposed in [25].

In addition to the ℓ^1 regularization term, a small ℓ^2 term is also added to overtake numerical problems that can results from using the second order method, leading to the so called *elastic-net* penalty [26]. Benefits of the *elastic-net* regularization are two-fold. It enables efficient features selection, without any loss in resulting model's quality. Moreover, the obtained models are interpretable, allowing for features contribution analysis. It should be noted that these advantages do not entail a change in the number of model's parameters, nor a higher computational complexity.

3.2 *Features*

The maximum entropy framework, along with ℓ^1-regularization allow for a wide marge of freedom when engineering features. The ones described in [6] are used in addition to features described here. Discretization of continuous features is performed in a pre-processing step, using an unsupervised equal frequency interval binning method. Fine-grained versions of all feature functions are added by conditioning on current POS tags. Learning a separate weight for each, allows the model to pay more or less attention to features depending on the related tags.

WORD FEATURES describe the source and target words associated with the given link. In addition to features described in [6] we include (1) *Lexical probability (WProb)* A separate feature for each discretized probability $p(f_i|e_j)$ and $p(e_j|f_i)$, produced by IBM model 1. (2) *Word frequency (WFreq)* The source and target word frequency (and their ratio) computed as the number of occurrences of the word form in training data. (3) *Lexical Prefix/Suffix (WPref,WSuff)* A separate feature for each prefix/suffix of a predefined length (and their concatenation), for $l_{i,j}$ source and target words. (4) *Word similarity (WSim)* These features reflects that proper nouns are likewise translated in different languages, e.g. "SdAm

Hsyn" [1] and "Saddam Hussein". A separate feature is defined per distinct value of the word similarity between $l_{i,j}$ source and target words. We employ the Levenshtein (edit) distance as a measure of similarity. (5) *Identity (WIdent)* is a binary feature which is active whenever f_i is equal to e_j (useful for untranslated numbers, symbols, names, and punctuation). (6) *Punctuation mismatch (WPunct)* indicates whenever a punctuation is aligned to a non-punctuation.

ALIGNMENT MATRIX FEATURES characterize the set of input alignment matrices, in addition to their union matrix $\mathbf{A}_{\cup}$. In addition to features described in [6] we include *multiple distortion (AMultd)* feature, which indicates if a link involves a duplicated word. Indeed, duplicated words could be misaligned due to a weak distortion model in comparison with lexical probabilities in IBM alignments [27]. E.g. several "fy" on the source side could be erroneously aligned to the same "in" on the target side regardless of the distortion. This feature is active for the link $l_{i,j}$ if f_i or e_j is duplicated, returning the distance to the diagonal.

4 EXPERIMENTAL RESULTS

All reported results have been obtained on the Arabic-English language pair, using data described in Table 1. The IBM Arabic-English manually aligned corpus (IBMAC) supplies our gold alignments. It includes the NIST MT Eval'03 as a test set, and a training set that we split into disjoint train and dev sets, used respectively for training and tuning our discriminative models. For ME training we used Wapiti [26][2], whereas the generative models are estimated using MGIZA++[3]. Default configurations are considered for the phrase based translation system Moses[4]. A 4-gram back-off language model, estimated with SRILM[5] is used in all our experiments. Minimum Error-Rate Training [28] is carried on to tune the parameters of the translation system. MADA+TOKAN[6] D2 tokenization scheme is used in a pre-processing step, to take Arabic rich

[1] All Arabic transliterations are provided in the Buckwalter transliteration scheme
[2] http://wapiti.limsi.fr/
[3] http://geek.kyloo.net/
[4] http://www.statmt.org/moses/
[5] http://www-speech.sri.com/projects/srilm/
[6] http://www1.ccls.columbia.edu/ cadim/MADA.html

Table 1. Experimental data: number of sentences and running words. G and D are acronyms for *generative* and *discriminative*, respectively.

Data source		#Sent	#Ar tok	#En tok	Usage
IBMAC	*test*	663	16K	19K	Evaluating all alignments
	dev	3,486	71K	89K	Tuning discriminative alignments
	train	10K	215K	269K	Training discriminative alignments
MT'08	*test*	1,360	43K	53K	Evaluating translations
MT'06	*test*	1,797	46K	55K	Tuning Moses parameters
MT'09	*constrained*	5M	165M	163M	Training MGIZA, SRILM and Moses

morphology into consideration. Inconsistency with tokenization of the IBMAC corpus is handled using the *splitting/remapping* technique described in [6]. We evaluate the quality of alignments compared to a gold standard using Alignment Error Rate (AER). The impact on translation quality is measured using multi-reference BLEU [29].

4.1 *Oracle study*

As explained in 3.1, limiting the search space to the union of input alignments, establishes an upper bound on the recall, preventing the model from reaching plausible links. In this oracle study, we quantify manual alignment reachability by several combination of input alignments, with different window sizes. Table 2 summarizes the percentage of alignment matrix covered by the union of input alignments, with its recall and AER according to the gold alignment. Oracle AER drops drastically when increasing the size of the window. Take, for instance, the case of IBM1 models: using a window of size 1 instead of 0 reduces the oracle by 10.8 points (from 13.7 to 2.9) at the cost of exploring larger area of alignment matrix (23.5% instead of 4.1%). It is worth noticing that the HMM model achieves similar oracle scores as IBM4, while its training and inference are fast and exact. Moreover, combining IBM1 and HMM results in comparable performances to the standard symmetrization heuristic (which has an oracle of 6.0 for the best IBM model), while exploring a slightly wider search space. Increasing the window size allows to largely outperform the heuristic with a much wider search space. This study suggests that most manual alignments are proposed by the input generative

Table 2. Search space coverage for different window sizes, and associated Oracle AER for different input alignments. W is the window size.

Input Alignments	Search Space %			Union Recall %			Oracle AER %		
	W=0	*W=1*	*W=2*	*W=0*	*W=1*	*W=2*	*W=0*	*W=1*	*W=2*
IBM1	4.1	23.5	43.9	75.9	94.3	98.7	13.7	2.9	0.7
HMM	3.3	15.9	26.9	85.3	97.0	98.7	7.9	1.5	0.6
IBM4	**3.3**	15.7	26.6	88.7	98.4	99.4	**6.0**	0.8	0.3
IBM1 + HMM	**5.0**	**25.4**	45.4	87.3	98.3	99.6	**6.8**	**0.8**	0.2
ALL	5.5	26.8	**47.0**	90.8	99.2	99.7	4.8	0.4	**0.1**

models, and justifies their use to prune the search space (an AER of 0.1 is achievable by examining 47% of the matrix).

4.2 *Impact of system components on AER*

The framework we consider for word alignment includes several interacting components. We design the following experiment, in order to quantify the contribution of each of them. We train a baseline system and measure its performance in terms of AER. We then train several models, in each of which, only one component differs from the baseline, and we compare their performance.

All the new features added over the baseline described in [6] help improving both recall and precision. They can be divided in two classes according to their discrimination power: first come WPreff, WSuff, WProb and WFreq with about 0.5 AER reduction each, then AMultd, WIdent, WPunct and WSim with about to 0.2 AER reduction each. Including all the new features improves AER by 1.6 over the baseline. Different thresholds α are employed to test different balance point between precision and recall. Thresholds between 0.1 and 0.9 shift precision from 81.9 to 92.7 and recall from 72.5 to 64.7. The lowest AER for the baseline (22.4) is achieved at $\alpha = 0.5$. The more annotated training data is used by the discriminative model, the better. Increasing the training corpus size from 10 sentences to 10K, enhances the AER by 7 points.

Results for the other components can be founds in Table 3. We note that aggressive features pruning with high values of the ℓ^1 regularizer, results in improved precision and recall (hence AER). The biggest AER reduction of 1.2 points (at $\ell^1 = 3$) is attainable while discarding 97% of the features. We note also that increasing the size of the search space,

Table 3. Only the component in the first columns changes with respect to the baseline. The left part of the table shows the impact of (1) different values for ℓ^1 regularization. While the right part shows the impact of (2) different search space controlled by the size of the window, (3) different input alignments quality determined by the size of their trainind data, and (4) stacking. Baseline: IBM1 input alignments, window w=0, $\ell^1 = 0$, $\ell^2 = 0.01$, threshold $alpha = 0.5$, 2k word-aligned sentence (to train the discriminative models) and 5M parallel sentence to train IBM input models.

Align	Pr	Rc	AER	# ftr
Baseline	87.9	69.4	22.4	501238
(1) ℓ^1 *regularization*				
ℓ^1=0.1	86.7	69.4	22.9	92590
ℓ^1=0.5	88.0	69.9	22.1	50380
ℓ^1=1	88.8	70.2	21.6	35268
ℓ^1=2	89.3	70.3	21.3	19610
ℓ^1=3	89.4	70.4	21.2	13806
ℓ^1=4	89.3	70.3	21.3	10704
ℓ^1=5	89.4	70.0	21.5	8528
ℓ^1=6	89.1	70.2	21.5	7334

Align	Pr	Rc	AER
(2) Search space: window size			
W=0	87.9	69.4	22.4
W=1	78.0	82.0	20.1
W=2	77.2	81.6	20.6
(3) Input alignment quality			
30K	85.9	64.0	26.7
130K	87.2	66.1	24.8
1030K	87.3	68.3	23.4
(4) Stacking			
Stack	89.1	70.2	21.4

using larger windows around the current link (e.g. w = 1), reduces the AER by 2.3. When exploring a wider search space, the model is able to retrieve more links, improving the recall by 12.6 points. But it makes more mistake, since it has to make more decisions, and hence degrade precision by 9.9 points. It should be mentioned subsampling methods to treat the imbalanced data problem related to wide search spaces did not help.

To evaluate the model's sensitivity to the quality of input alignments, we exploit the fact that training MGIZA alignments with less data results in alignments with degraded quality: we train IBM model 1 with MGIZA using corpora of different sizes (30K, 130K, 1030K). Each on of these alignments is then used as an input to build a discriminative system. The resulting systems are then compared to the baseline, which is build using IBM model 1 alignment trained on the entire 5M parallel corpus.

The baseline's AER drops from 22.4 to 26.7 for the worst input alignment (IBM1 trained on 30K). Stacking helps correcting errors in the baseline and improve its AER by 1 point by enhancing both recall and precision.

Table 4. Sample of selected features with high weights

Feature	Weight
$l_{i,j} = active \wedge WPref(f_i) = Al\$ \wedge WPref(e_j) = el-$	1.7313
$l_{i,j} = active \wedge WPref(f_i) = Anh \wedge WPref(e_j) = tha$	1.6652
$l_{i,j} = active \wedge POS(f_i) = CC \wedge POS(e_j) = CC$	1.4559
$l_{i,j} = inactive \wedge WPunc(f_i, e_j)$	1.2070
$l_{i,j} = active \wedge MGIZA\_HMM(f_i, e_j) = active$	0.7639

4.3 *Model and features selection*

As described in section 3.1, the use of ℓ^1 regularization yields a sparse model where the most useful features have been selected during the training step. Some of these features are shown in Table 4: the first binary feature indicates if the Arabic word starts with the prefix *Al* while the English word begins with the *el* prefix. This feature indeed embeds a rule of thumb to translate Arabic proper noun, and is sufficient to ensure correct alignments for all the related occurrences in the test set. The second feature encodes the punctuation mismatch and prevents to align punctuations with regular words. With this feature, the model prefers to leave a punctuation not aligned, rather than aligned with a regular word. This decision is generally the best if a punctuation cannot be aligned with another punctuation.

Even if most of the selected features are related to the input generative models HMM and IBM1 (40% of the features), a more global study shows that all classes are represented in the final model and so are useful for alignment. Moreover, it is worth noticing that 90% of the selected features are those conditioned on current POS tags.

4.4 *Alignments characteristics and BLEU*

In these experiments, we aim to assess the quality of the new alignments measured by AER, and their impact on the translation quality measured by BLEU. We use two different baselines: generative IBM alignments and discriminative ME alignments described in [6]. In Phrase Based Statistical Machine Translation (PBSMT), the basic translation unit is a phrase pair and its associated scores. Phrase pairs are extracted from a parallel corpus to build the phrase table which contains ideally all sub-sentential translational equivalences. The quality of these phrase

pairs is determined by the number of correct translational correspondences they can capture. In practice, word-level correspondences are precomputed, then fed to an extraction heuristic that generalizes these correspondences to the phrase level. In this scenario, two sources of errors may affect phrase pairs consistency. On the one hand, word alignments are error prone, and they fail sometimes to detect word-level translations which carry on to the extracted phrase pairs. However, the extraction heuristic achieves generalization by combining aligned words into phrases, and growing over unaligned ones around them. This can be helpful to treat cases where no word-level alignment exists, such as in the translation of propositions, idiomatic expressions and compound words. However, since this heuristic operates locally on a sentence level and makes heursitic decisions, it can easily extract noisy phrases, especially when given a wide margin of freedom by leaving many words unaligned.

Since word alignments are the only constraints on the extraction heuristic, they become the only way to control both sources of errors mentioned earlier: by settling on a good balance between the alignment quality and the number of unaligned words. Therefore, the tradeoff between precision and recall for word alignments has a great impact on the quality of the extracted phrases. For instance, let us consider the case of a perfect precision (all links are correct), but with a low recall (not all word-level correspondences are detected). Then the alignment matrix is sparser than it should be, and the proportion of unaligned words results in many phrase pairs, with moderate scores (since they allow for multiple translations which over-flatten the probability distribution). Human-perceived quality of resulting phrases also degrades [7]. In the other case, with a high recall and low precision, the alignment matrix is denser than it should be, and generalization fails, with fewer and over-deterministic phrase pairs. Thus, the quality of a phrase table depends on the interaction between the quality of word alignments (precision and recall) and the sparsity of the alignment matrix: the number of unaligned source or target words, and the resulting gaps.

Our results are interpretable in light of this discussion. IBM1 is an example of alignments were both sources of errors are apparently affecting its performance. Compared to a manual alignment: IBM1 (1) produces poor alignment quality with low precision and recall, which causes the extraction of erroneous phrase pairs, and (2) leaves more words unaligned, which adds to the noise in extracted phrases.

Subsequent generative models are more efficient: HMM and IBM4 improves both precision and recall, while aligning more words. These

Table 5. Characteristics of alignments in terms of their quality compared to gold standard, number of links and unaligned source/target words. Number of extracted phrases is included with average number of gap per source/target word, and percentage of gapless phrases. These statistics are calculated using the IBMAC test set 1. Finally the quality of alignments in terms of their impact on BLEU for three different MT tasks. Th is the threshold.

Alignment Characteristics						**Phrase-pairs Characteristics**					**BLEU**		
Recall	***Precision***	***AER***	***#links***	***#UnalignSrc***	***#UnalignTgt***	***#Phrases***	***avgSrcGap***	***%GaplessSrc***	***avgTgtGap***	***%GaplessTgt***	***30K***	***130K***	***1030K***
Manual alignments													
100	100	0.0	16171	2655	3457	86642	0.68	56.5	0.83	47.6	-	-	-
Generative baselines (gdfa): *IBM1, HMM, IBM4*													
70.2	71.0	29.4	16394	3032	4752	72369	0.98	47.6	1.28	36.9	36.0	39.2	40.6
73.7	81.4	22.6	17985	1967	3524	74782	0.63	62.8	0.96	46.6	37.5	40.5	41.5
75.1	86.1	**19.8**	18715	1422	2460	60029	0.34	75.4	0.57	61.0	38.0	41.1	**42.0**
Discriminative baseline [6]: IBM1-HMM ($th = 0.6$), +Stacking ($th = 0.5$)													
90.7	82.0	13.9	14733	3435	4851	119303	1.04	44.2	1.29	33.9	38.0	41.3	42.3
92.7	81.5	**13.2**	14953	3371	4542	122412	0.99	44.8	1.13	34.4	38.2	41.4	**42.3**
New system: *IBM1-HMM ($th = 0.5$), +Window ($th = 0.8$), +Stacking ($th = 0.7$)*													
91.4	82.7	13.2	15215	3197	4552	106490	0.93	47.5	1.18	36.8	38.2	41.4	42.8
89.7	86.6	11.8	17143	3436	4019	107063	1.05	43.7	1.14	39.2	37.4	40.5	41.8
93.1	86.5	**11.2**	16054	3008	4173	108825	0.91	46.6	1.15	38.9	38.5	41.7	**42.9**

enhancements lead to the extraction of less noisy phrase pairs, which perform better. IBM4 for example, improves BLEU by 2 points over IBM1 for the smallest task, and 1.4 points for the biggest one.

Discriminative baseline alignments have different profile than IBM models. They are more efficient in retrieving more correct links, as well as greatly improving recall (from 75% for IBM4 to 92.7% for the stacked baseline). Thus, the quality of extracted phrase pairs should be improved significantly since they are based on better word-level correspondences, and the first source of errors is limited. But on the other hand, these alignments tend to be sparser than gold standards (9% less links) and IBM4 model (21% less links), which causes more extraction errors, degrading back the quality of phrase pairs. Otherwise stated, the improvement in

phrase table quality due to AER improvement, is almost cancelled out by increasing the percentage of gaps. Which explains why discriminative baselines achieve small improvements over IBM models.

The new system has a profile similar, in general, to the discriminative baseline: improved AER, and sparser alignments. The first line in the new system part of the table 5 describes the first new system, which uses the better feature engineering and ℓ^1 regularization (without the enhanced search space). It achieves comparable alignment quality (precision, recall) to the discriminative baseline, but it is able to align more words, and decrease the percentage of gaps. This results in higher BLEU scores on all the three tasks.

Adding the enhanced search space (window of size 1) to the previous system, allows for a significant increase in recall (from 82.7% to 86.6%) with slightly degraded precision, which improves the AER. These alignments change the balance between unaligned source and target words, with respect to the previous system: more *source* words, and *less* target words are aligned, in a comparable sized phrase table. This configuration is harmful and results in about 1 BLEU point loss on all tasks. An interesting result comes in the next line, when adding a stacking layer to the system with the enhanced search space. Stacking fixes the problem with precision, without harming recall, improves the over all quality of the alignment, and reduces the number of unaligned source words, shifting the balance back. This system achieves the lowest alignment error rate of 11.2%, and the best BLEU score on all three tasks, with significant improvements over the generative and discriminative baselines (for the biggest task).

5 CONCLUSIONS

In this paper we introduced an improved discriminative alignment system, that is based on a maximum entropy framework for combining word alignments. Better feature engineering, combined with ℓ^1 regularization and an enhanced search space allow the model to improve both the quality of alignment and translation. The introduced model achieves an overall reduction of 43% of the alignment error rate over the standard IBM model 4 symmetrized alignment, resulting in 0.9 point enhancement in BLEU. While adding a stacked classification layer may not be very helpful to the discriminative baseline introduced in [6], it proves to be necessary to allow the new system to benefit from the enhanced search space and achieve improvements in translation quality.

We analyzed the BLEU results in light of several alignment characteristics and noticed that finding a better balance between the alignment quality measured by its precision and recall, its sparsity, and its number of unaligned words and extracted phrases is necessary to deliver better translation models.

ACKNOWLEDGMENTS The work was partly realized as part of the Quaero Program, funded by OSEO, the French agency for innovation.

REFERENCES

1. DeNero, J., Klein, D.: The complexity of phrase alignment problems. In: HLT. (2008) 25–28
2. Brown, P.F., Pietra, S.A.D., Pietra, V.J.D., Mercer, R.L.: The mathematics of statistical machine translation: Parameter estimation. Comput. Linguist. **19** (1993) 263–311
3. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: NAACL. (2003) 48–54
4. Ayan, N.F., Dorr, B.J.: A maximum entropy approach to combining word alignments. In: HLT-NAACL. (2006) 96–103
5. Elming, J., Habash, N.: Combination of statistical word alignments based on multiple preprocessing schemes. In: NAACL-HLT. (2007) 25–28
6. Tomeh, N., Allauzen, A., Yvon, F., Wisniewski, G.: Refining word alignment with discriminative training. In: AMTA. (2010)
7. Guzman, F., Gao, Q., Vogel, S.: Reassessment of the role of phrase extraction. In: 12th MT Summit. (2009)
8. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29** (2003) 19–51
9. Blunsom, P., Cohn, T.: Discriminative word alignment with conditional random fields. In: ICCL and ACL. (2006) 65–72
10. Matusov, E., Zens, R., Ney, H.: Symmetric word alignments for statistical machine translation. In: COLING. (2004)
11. Wu, D.: Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics **23** (1997) 377–403
12. Tiedemann, J.: Word to word alignment strategies. In: COLING. (2004) 212–218
13. Lin, D., Cherry, C.: Word alignment with cohesion constraint. In: HLT-NAACL. (2003)
14. Zhao, B., Vogel, S.: Word alignment based on bilingual bracketing. In: HLT-NAACL 2003. (2003) 15–18
15. Cherry, C., Lin, D.: A probability model to improve word alignment. In: ACL. (2003) 88–95

16. Ittycheriah, A., Roukos, S.: A maximum entropy word aligner for Arabic-English machine translation. In: HLT '05. (2005) 89–96
17. Liu, Y., Liu, Q., Lin, S.: Log-linear models for word alignment. In: ACL. (2005) 459–466
18. Niehues, J., Vogel, S.: Discriminative word alignment via alignment matrix modeling. In: Proc. of the 3rd Workshop on SMT. (2008) 18–25
19. Moore, R.C.: A discriminative framework for bilingual word alignment. In: HLT. (2005) 81–88
20. Taskar, B., Lacoste-Julien, S., Klein, D.: A discriminative matching approach to word alignment. In: HLT '05. (2005) 73–80
21. Lacoste-Julien, S., Taskar, B., Klein, D., Jordan, M.I.: Word alignment via quadratic assignment. In: NAACL-HLT. (2006) 112–119
22. Wolpert, D.H.: Original contribution: Stacked generalization. Neural Netw. **5** (1992) 241–259
23. Cohen, W.W., Carvalho, V.R.: Stacked sequential learning. In: IJCAI. (2005) 671–676
24. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Statist. Soc. B **58** (1996) 267–288
25. Andrew, G., Gao, J.: Scalable training of L1-regularized log-linear models. In: ICML. (2007) 33–40
26. Lavergne, T., Cappé, O., Yvon, F.: Practical very large scale CRFs. In: ACL. (2010)
27. Riesa, J., Marcu, D.: Hierarchical search for word alignment. In: ACL. (2010)
28. Och, F.J.: Minimum error rate training in statistical machine translation. In: ACL. (2003) 160–167
29. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL. (2002) 311–318

NADI TOMEH
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <NADI.TOMEH@LIMSI.FR>

ALEXANDRE ALLAUZEN
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <ALEXANDRE.ALLAUZEN@LIMSI.FR>

THOMAS LAVERGNE
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <THOMAS.LAVERGNE@LIMSI.FR>

FRANÇOIS YVON
LIMSI/CNRS AND UNIV. PARIS SUD,
BP 133, 91403 ORSAY CEDEX,
FRANCE
E-MAIL: <FRANCOIS.YVON@LIMSI.FR>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 257–263
RECEIVED 01/10/11 ACCEPTED 15/11/10 FINAL 16/12/11

A New Combined Lexical and Statistical based Sentence Level Alignment Algorithm for Parallel Texts

GRIGORI SIDOROV,[1] JUAN-PABLO POSADAS-DURÁN,[1]
HECTOR JIMÉNEZ-SALAZAR,[2] AND LILIANA CHANONA-HERNANDEZ[1]

[1] *National Polytechnic Institute (IPN), Mexico*
[2] *Autonomous University of Mexico State (UAEM), Mexico*

ABSTRACT

Parallel texts alignment is an active research area in Natural Language Processing field. In this paper, we propose a method for sentence alignment of parallel texts that is based both on lexical and statistical information. The alignment procedure uses dynamic programming technique. We made our experiments for Spanish and English texts. We use lexical information from bilingual Spanish-English dictionary, as well as the sentence length measured in words and in characters. The proposed method was tested on a corpus of fiction texts, where the frequency of multiple alignments, omissions and insertions is higher than in other types of texts. We obtained better results than the standard Vanilla aligner system that uses a purely statistical approach.

KEY WORDS: *Parallel texts alignment, English, Spanish, dynamic programming, lexical-based alignment, statistical alignment, anchor points, sentence length, fiction writing.*

1 INTRODUCTION

Parallel texts alignment is an active area of research in Natural Language Processing field. Parallel texts are translations of each other

in different languages. The works in this area started at wide scale in 1990s [1, 2, 4, 7, 8]. Some modern ideas of optimization, for example, with genetic algorithms [5], and the task of alignment of more challenging types of texts (like fiction) [6] maintain the interest to this topic.

The usefulness of the aligned parallel texts is related to the fact that they explicitly contain the relationship between the elements in a text in one language and elements of the same text translated into another language, thus, allowing performing of numerous tasks that can exploit the knowledge of the alignment. For example, alignments are useful for machine learning and automatic extraction of information for various purposes, such as statistical machine translation, bilingual word sense disambiguation, etc.

In this work, we propose a method for sentence alignment in parallel texts. We tested it for Spanish and English language pair, though the algorithm is language-independent. It is based on both lexical and statistical information and uses dynamic programming framework [6]. The lexical information is contained in a bilingual dictionary, while statistical information is obtained from the sentence lengths like in [4] measured both in words and in characters.

The proposed method was tested on a corpus of fiction texts. Note that fiction texts are much more difficult case for alignment: the frequency of multiple alignments, omissions and insertions is higher than in texts of other types, like technical or law texts.

We obtained precision about 96%. We compared our results with the results obtained by the Vanilla aligner system [3] for the same corpus. Note that Vanilla uses a purely statistical approach. Vanilla obtained precision about 92%.

The developed method is superior in cases of multiple alignments, omissions and insertions. The results that we obtained show that the use of lexical information contained in a bilingual dictionary combined with statistical information allows developing of the robust method for sentence alignment in fiction texts, which gives better results than purely statistical methods.

The paper is organized as follows. First we describe the alignment algorithm, then we present the corpus used in the experiments and after this present and discuss the obtained results, finally, conclusions are drawn.

2 PROPOSED ALGORITHM

In the task of sentence alignment, we need to find the correspondences of sentences in one text and the sentences of its translation. This correspondence is not trivial because some sentences can be omitted and some sentences can be translated by two or more sentences. The task of sentence level alignment is well-known and intuitively clear, we send the reader for details to the works [4, 6].

We use dynamic programming approach to find the best alignment for all sentences in a pair of texts [4, 6]. Note that this approach implies continuity, i.e., if we found an alignment of a sentence, then no posterior sentence can be aligned before the already existing alignment.

In this approach, we assign scores to every possible pair of aligned sentences. The final score of the alignment is the sum of the scores of each sentence pair that constitutes this alignment.

For weighting the similarity, we use the concept of *significant elements* that are words of open POS classes: nouns, verbs, adjectives, adverbs, and also proper names, abbreviations, signs of admiration, signs of interrogation, and numbers.

We use the following equation for calculating the similarity between sentences. In fact, this function is dissimilarity, i.e., penalization, but penalization is "inverse" function to similarity. Thus, the lower this value, the better is the similarity of the sentences.

$$\begin{aligned} \mathit{Similarity}\,(S_S; S_T) = {} & \mathit{DictionaryDiff}\,(S_S; S_T) + \\ & \mathit{SignificantElementsDiff}\,(S_S; S_T) + \\ & \mathit{CharLengthDiff}\,(S_S; S_T) \end{aligned} \quad (1)$$

where S_S is a source sentence, S_T is a target sentence, the value *SignificantElementsDiff* $(S_S; S_T)$ is the absolute value of the difference of number of significant elements, and *CharLengthDiff* $(S_S; S_T)$ obtains the absolute value of the difference of numbers of characters in two sentences, and *DictionaryDiff* is calculated as follows.

$$\begin{aligned} \mathit{DictionaryDiff}\,(S_S; S_T) = {} & \mathit{SignificantElements}\,(S_S) + \\ & \mathit{SignificantElements}\,(S_T) - \\ & 2 \times \mathit{Translations}(S_S; S_T) \end{aligned} \quad (2)$$

Thus, *DictionaryDiff* $(S_S; S_T)$ is the number of significant elements that are not mutual translations. In our experiments, we used Vox dictionary available on the Web.

```
<Preproccessing>
  <text file="holmes25_eng.txt" num_sentence="27" num_char="2703" language="English">
    <sentence id_sentence="1" num_words="4" mean_words="4" char="19" language="English">
      <token char="Sir" class="normal">
        <type char="sir">
          <translation>señor</translation>
          <translation>sir</translation>
        </type>
      </token>
      <token char="Arthur" class="nombre_propio">
        <type char="arthur">
          <translation>arthur</translation>
        </type>
      </token>
      <token char="Conan" class="nombre_propio">
        <type char="conan">
          <translation>conan</translation>
        </type>
      </token>
      <token char="Doyle" class="nombre_propio">
        <type char="doyle">
          <translation>doyle</translation>
        </type>
      </token>
    </sentence>
    <sentence id_sentence="2" num_words="5" mean_words="3" char="29" language="English">
      <token char="The" class="palabra_auxiliar">
        <type char="the"/>
      </token>
      <token char="adventures" class="normal">
        <type char="adventure">
          <translation>aventura</translation>
```

Fig. 1. Text representation using XML scheme.

In addition, we use anchor points as a constraints in the alignment, i.e., the alignment is possible only between established anchor points.

We used XML scheme of representation of the text, see Fig. 1. Traditional preprocessing with morphological normalization was performed.

3 Corpus Description

We used the following corpus for our experiments: Four chapters of "Adventures of Sherlock Holmes" by A. Conan Doyle and two chapters of "Turn of the screw" by Henry James. The text by A. Conan-Doyle contains 2,558 sentences in English and 2,267 sentences in Spanish. The text by Henry James contains 361 sentences in English and 363 sentences in Spanish. Corpus characteristics related to number of different types of alignments are presented in Table 1.

Manual alignment was performed for these texts, thus, representing the golden standard for the evaluation.

Table 1. Corpus characteristics: numbers of different types of alignments.

Alignment type	A. Conan-Doyle	H. James
1:1	2,061	300
1:2	65	12
2:1	64	8
2:2	4	1
3:1	1	0
1:3	2	0
1:0	11	1
0:1	7	1

4 Obtained Results and Discussion

We compared on the developed corpus our method and Vanilla aligner. Vanilla aligner [3] is based on the work of Gale and Church [4] and takes into consideration the following alignment types between sentences 1:1, 1:0, 0:1, 1:2, 2:1 and 2:2. The system uses the same input as our system, i.e., the text is splitted into sentences in the same way. It is purely statistical method based on lengths of sentences and dynamic programming.

We measure the precision of both methods as the percentage of correctly aligned sentences as compared to the golden standard. We used complete evaluation, i.e., if only a part of a multiple alignment is correct, we do not consider such alignment as the correct one.

The results of comparison of our algorithm with Vanilla aligner for the texts of Henry James and A. Conan-Doyle are presented in Table 2.

Table 2. Comparison of the results.

	Henry James	A. Conan-Doyle
Vanilla aligner	93.01%	90.66%
Proposed algorithm	**96.78**%	**95.62**%

The reasons of errors of our method are related with two major phenomena. The first one is that the word cannot be found in the bilingual dictionary, either due to incompleteness of the morphological analysis system, or due to the incompleteness of the bilingual dictionary itself.

The other reason is related to idiomatic expressions, like "*Oh, dear*" that is translated as "*Válgame Díos*" (lit. "*God help me*)".

5 Conclusions

The aim of this work was to propose an alignment algorithm that is based on combination of lexical information obtained from a bilingual dictionary and traditional statistical information related to sentence lengths.

We tested it for a special type of parallel texts that constitutes more complicated case of alignment, namely, fiction texts. Our hypothesis was that for these texts statistical aligner will make mistakes that can be corrected using lexical information.

We conducted our experiments on a relatively large corpus of fragments of works of A. Conan-Doyle and Henry James in Spanish and in English.

We obtained better results than the base line Vanilla aligner software. Our precision was about 96%, while Vanilla aligner obtained about 92%. The difference is due to better performance of our method for cases of one to many alignments, insertions and omissions.

Acknowledgements**.** Work done under partial support of Mexican Government (CONACYT projects 50206-H and 83270, SNI), Government of Mexico City (project PICCO10-120), European project 269180 WIQ-EI, project "Answer Validation through Textual Entailment" CONACYT-DST (India), and National Polytechnic Institute, Mexico (projects SIP 20111146, 20113295; PIFI, COFAA).

References

1. Brown, P., Lai, J., Mercer, R.: Aligning sentences in parallel corpora. In: Proceedings 29ht Annual Meeting of the ACL, 1991, pp. 169−176.
2. Chen, S.F.: Aligning sentences in bilingual corpora using lexical information. In: Proceedings of ACL-93, 1993, pp. 9−16.
3. Danielsson, P., Riddings, D.: Practical presentation of a Vanilla aligner. In: TELRI Workshop in Alignment and Exploitation of Texts, 1994, pp. 1−2.

4. Gale, W.A., Church, K.W.: A program for aligning sentences in bilingual corpora. In: Proceedings of the 29th annual meeting on Association for Computational Linguistics, 1991, pp. 177−184.
5. Gautam, M., Sinha, R.: A hybrid approach to sentence alignment using genetic algorithm. In: Proc. of International Conference on Computing: Theory and Applications, 2007, pp. 480−484.
6. Gelbukh, A., Sidorov, G.: Alignment of paragraphs in bilingual texts using bilingual dictionaries and dynamic programming. Lecture Notes in Computer Science, vol. 4225, Springer-Verlag, pp. 824−833.
7. Kay, M., Roscheisen, M.: Text-translation alignment. Computational Linguistics, 19(1), 121−142, 1993.
8. Simard, M., Foster, G., Isabelle, P.: Using cognates to align sentences in bilingual corpora. In: Proc. of TMI, 1992, pp. 67−81.
9. Xiong, H., Xu, W., Mi, H., Liu, Y., Liu, Q.: Sub-sentence division for tree-based machine translation. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, ACLShort'09, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 137−140.

GRIGORI SIDOROV
CENTER FOR COMPUTING RESEARCH (CIC),
NATIONAL POLYTECHNIC INSTITUTE (IPN),
COL. NUEVA INDUSTRIAL VALLEJO, CP 07738, DF, MEXICO
E-MAIL: < SIDOROV @ CIC.IPN.MX>

JUAN-PABLO POSADAS-DURÁN
CENTER FOR COMPUTING RESEARCH (CIC),
NATIONAL POLYTECHNIC INSTITUTE (IPN),
COL. NUEVA INDUSTRIAL VALLEJO, CP 07738, DF, MEXICO
E-MAIL: < JPDURAN @ CIC.IPN.MX>

HECTOR JIMÉNEZ-SALAZAR
AUTONOMOUS UNIVERSITY OF MEXICO (UAM),
UNIDAD CUAJIMALPA, DF, MEXICO
E-MAIL: < HGIMENEZS @ GMAIL.COM>

LILIANA CHANONA-HERNANDEZ
ENGINEERING FACULTY (ESIME),
NATIONAL POLYTECHNIC INSTITUTE (IPN),
COL. ZACATENCO, CP 07738, DF, MEXICO
E-MAIL: < LCHANONA @ GMAIL.COM>

Applications

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 267–283
RECEIVED 02/11/10 ACCEPTED 26/11/10 FINAL 11/04/11

Classification of Attitude Words for Opinion Mining

LARITZA HERNÁNDEZ,[1] A. LÓPEZ-LOPEZ,[1] AND JOSÉ E. MEDINA[2]

[1] *Instituto Nacional de Astrofísica, Óptica y Electrónica, México*
[2] *Advanced Technologies Application Center, Cuba*

ABSTRACT

This work details appraisal extraction from attitude expressions. Here, by attitude expressions, we refer to those single words that convey the evaluation of sentiments or emotional states, about human behaviors, objects, processes or people, according to the Appraisal Theory of language. The attitude words can be classified into affect, judgment, and appreciation; either positive or negative. Extraction of the attitude words has a significant range of applications from opinion extraction and summarization, up to temporal opinion analysis. To determine the attitude, we use two machine learning techniques; namely, Support Vector Machines and Random Forest. These algorithms classify a given word starting from a vector that represents the information from the context where the words tend to occur. On the other hand, we can observe the context of the words relying on a corpus of sentences from user generated contents, such as reviews, editorials and other online texts.

KEY WORDS: *Opinion Extraction, Appraisal Theory, Corpus Evaluation, Machine Learning.*

1 INTRODUCTION

Evaluation, according to Hunston and Thompson in 2000, "*is the broad cover term for the expression of the speaker or writer's attitude or*

stance towards, viewpoint, or feelings about the entities or propositions that he or she is talking about" [1]. Appraisal Theory tries to explain the semantic option schemes that the language has to evaluate. It is actually focused on the linguistic expression of attitude, and it separates evaluation into three subsystems; namely, Attitude, Graduation and Engagement[1]. Attitude corresponds to the words that emit an evaluation or that invite to do it. Graduation considers the words that intensify, diminish, sharpen or blur the evaluation. Engagement corresponds to those words that indicate the posture that the issuer adopts with the statement.

The problem we try to solve in this paper (appraisal extraction from single words) is part of a more complex problem (appraisal extraction from phrases) which is our aim in future work. Therefore, here, we focus the Appraisal extraction only on Attitude from single words. When we will extend our work to phrases (word sequences), we will study the graduation and engagement components.

Appraisal Theory subdivides Attitude into affect (evaluation of sentiments or emotional states), judgment (evaluation of the human behavior), and appreciation (evaluation of objects, processes, or people when they are valued from an aesthetic viewpoint). Attitude, also, can be positive or negative.

We are mainly interested in recognizing appraisal on the Spanish language; since we have found few advances of Opinion Mining on this language. For this reason, we have prepared a word list of attitude; as well as a corpus of sentences in which is possible to observe the context of these words. Nevertheless, we consider that the assumptions in this paper can be applied to other languages. In Fig. 1, we can see some examples of the appraisal systems.

Extraction of the attitude words has a significant range of applications from opinion extraction and summarization [6], up to temporal opinion analysis [7]. Nowadays, many people express their sentiments, evaluations, or judgments online in a variety of sources, such as customer reviews, editorials, blogs, and others. Summarizing those opinions can be very helpful, but this requires the extraction of attitude key phrases that are cues of the opinions expressed in the text.

For example, in sentence (1) we can notice two subjective words, both with a positive polarity. We can infer that the polarity of the

1 Additional details about Appraisal Theory can be found in [2–5].

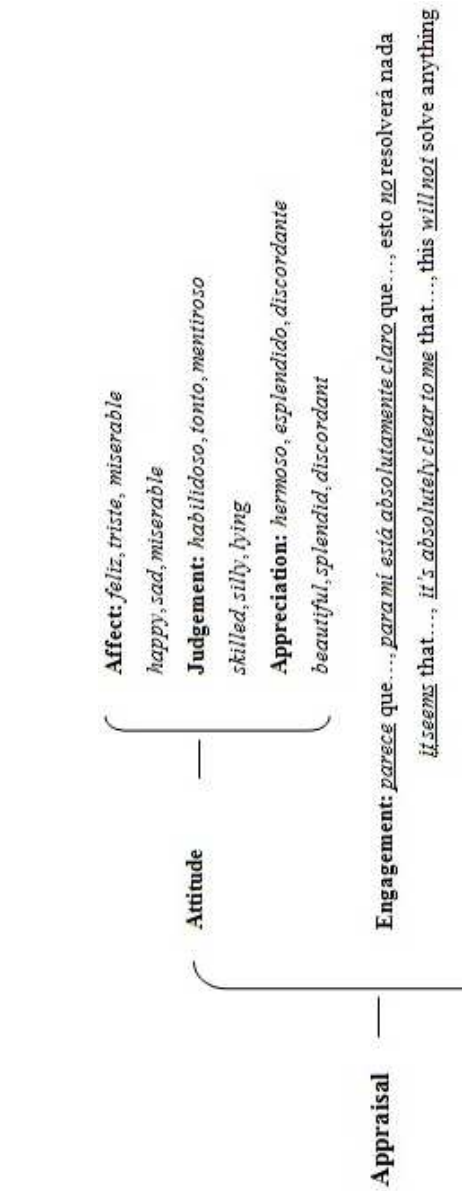


Fig. 1. Categories of Appraisal Theory and example words of Attitude system.

sentence is positive as well, by computing the amount of positive words[2].

(1) "*Viéndola, me doy cuenta de que si tanto me ha gustado*$^{+}$ *es porque la trama es comprensible*$^{+}$."

However, recognizing attitude in words (see sentence (2)) allows knowing the evaluation purpose of the sentence (sentiments, objects, or human behavior). Thus, sentences more relevant to a particular interest could be identified. For example, if the concerned item is the human behavior, you could be interested in summarize what anyone says about the capacity, or moral integrity of a given person, more than in its physical appearance or simple polarity.

(2) "*Viéndola, me doy cuenta de que si tanto me ha* [**affect**: *gustado*$^{+}$] *es porque la trama es* [**appreciation:** *comprensible*$^{+}$]."

On the other hand, the opinion usually tends to change on time, and the attitude extraction could be useful for monitoring or analyzing tendencies of public relations and marketing firms, opinions about products, people, organizations, etc.

Many words of an attitude system, according to the Appraisal Theory, have potential to express affect, judgment and appreciation when we consider them out of context, since affect is considered as the basic system of Attitude, whereas judgment and appreciation are derivations of this, manifesting institutionalized emotions. This situation has motivated us to use a corpus-based approach. This approach allows recognizing the evaluation of words considering the context where these tend to occur.

The present paper is structured as follows. In Section 2, we briefly explain the strategy to recognize appraisal in words. In Section 3, the structure and corpus composition are presented. In the last sections, we show the experimental results, as well as discussion and conclusions.

[2] Sentences that are shown as examples in this article are original sentences from the corpus we are working with, so we do not translate them here. Nevertheless, a translation for the sentence above is: *Seeing it, I realize that, if I liked it so much is because the plot is comprehensible.*

2 RECOGNIZING APPRAISAL

Appraisal extraction is a Sentiment Analysis task; this is a novel research area. Some initial works date back to the late 1980's and early 1990's [8], [9]; Sentiment Analysis is conceived as Sentiment Classification, referring to the task of categorizing texts, or pieces of text, based on their subjectivity and orientation [10]. Others extend it to identify or classify appraisal targets, determining the source of an opinion in a text, and developing interactive and visual opinion mining methods [11]. Sentiment classification has been mainly focused in polarity classification; i.e., it determines if appraisal is positive, negative, neutral or if there is no appraisal in the text.

2.1 RELATED WORKS

Many previous works in sentiment classification have shown good performance using a lexicon-based approach. Starting by word lists manually annotated, they classify a larger piece of text, such as sentences [12] and paragraphs [13]. On the other hand, most of the works are focused on the English language; we have only found a few proposals for Spanish language that try to solve this problem generating a subjectivity-annotated corpus or dictionaries from translation of English subjective texts. (Banea et. al. 2008) propose to annotate (subjective and objective) sentences in Spanish and Romanian, by employing machine translation and leveraging on the resources and tools available for English like MPQA corpus and Opinion Finder system, respectively [14]. (Bautin et. al. 2008) determine entity sentiment scores on nine languages of news sources and five languages of a parallel corpus; i.e., they calculate the polarity (positive and negative) and subjectivity score (how much sentiment of any polarity the entity receives) for a given entity, by using the automatic English translation of this languages and Lydia system for Sentiment score calculation [15]. (Brooke et. al. 2009) intend the creation of Spanish dictionaries, making an analogy with adjective, noun, verb, adverb, and intensifiers dictionaries in English [16]. Each adjective, noun, verb, adverb dictionary in English is automatically translated to Spanish by means of the online bilingual dictionary Spanishdict and online Google translator, maintaining the polarity of words from English. Also for the bilingual dictionary and translator, the author proposed other method

using a textual corpus in Spanish formed by 400 reviews about hotels, movies, music, phones, washing machines, books, cars, and computers. From this corpus, adjectives, nouns, verbs, adverbs, and intensifiers dictionaries were extracted, along the polarity for each word. That is a valid approach to cope with the problem of sentiment classification in Spanish. But, since the words "subjective sense", as well as the intensity of this subjectivity, can be lost in the translation, we consider that a more detailed study has to be done, where the particularities of this language are taken into account, avoiding loss of generality, as far as possible. On the other hand, although lexicon-based approach has shown good performance in sentiment classification, we consider that the dictionaries are not exhaustive, and an automatic classification of words can be more useful because it allows finding new appraisal words.

In a previous work about automatic word classification, Turney and Littman intend to infer the polarity a word from extremely large corpora, considering its semantic association with other words, which they called "paradigms"[17], [18]. That is, they use two lists of words, called *positive paradigms* and *negative paradigms*, and calculated the association probabilities of a given word with the paradigms as the number of returned matching documents from AltaVista Advance Search, by means of hits (query), and using the NEAR operator. This method depends on the variations and availability of an online search system. Besides, the NEAR operator considers that two words are close when they are halfway at least ten words, but it does not distinguish if the words are in the same sentence, an aspect that we consider important to consider.

Other work closer to our study that intends to classify adjectives in affect, judgement and appreciation was proposed by Taboada & Grieve work in 2004 [19]. They, with similar approach to Turney and Littman, use the NEAR operator on AltaVista Advance Search. Nevertheless they associated the word with a "pronoun-copula" pair (¨I was¨ for affect, ¨he was¨ for judgement, and ¨it was¨ for appreciation) instead of paradigms. This is a first interesting approach to classify attitude using context; but, there are several examples that show that the three proposed combinations (I was (affect), He was (judgment), ¨It was¨ (appreciation)) are not enough and they even fail in some cases.

2.2 OUR PROPOSAL

We proposed a strategy to distinguish words that convey appraisal of an item from the rest, as well as to classify the evaluation polarity (positive or negative). In addition, relying on Appraisal Theory, we classified the evaluation words into affect, judgment, and appreciation. Both, polarity and attitude are recognized using a corpus-based approach. This approach allows recognizing the attitude and polarity of the words determined by the context where they tend to appear.

As we know, many words in the human language are ambiguous (they do not convey a single message) when they are studied out of context; i.e., the context strongly determines the word sense. The evaluative language is not an exception (e.g. it is difficult to know if *big* or *much* conveys a negative or positive evaluation). On the other hand, according to proponents of the Appraisal Theory, some words out of context can be ambiguous according to their attitude class (e.g. *aburrido* (boring), *cómodo* (pleasant), or *agradable* (nice)).

First, we assume that the sentences are the smallest units of coherent messages in texts. Therefore, we assume that the words that tend to co-occur in the same sentences are used with the intention of expressing similar or identical messages. We only focus on the appraisal that is indicated in an explicit and direct way. We describe a supervised strategy to learn sentiment classifiers of words.

Then, considering the previous assumptions, we also assume that words with a given polarity probably tend to occur in sentences of the same polarity. That probably does not happen in sentences with different polarity or without polarity. Therefore, if we start collecting a set of seed words and its polarity (positive, negative, and no-polarity), and if we represent them by a vector of words that occur in their sentences; then we hypothesize that it is possible to learn the context (words) of each polarity class, increasing our lexicon with new words of the same polarity. We have noted that in the current work, we are only interested in determining the positive and negative categories.

Subsequently, the attitude class of words is related to its sense and to the item (sentiment, human behavior, or object) target of the evaluation. We had assumed that in a single sentence, the evaluation of a single item prevails. Therefore, we start from the hypothesis that the words that tend to co-occur in the same sentence are being used with the intention of expressing the same kind of attitude. In a previous work, we represent the words of attitude by a vector of dimension n, where n was the corpus size (sentences), assuming that sentences can

be adequate to discriminate the attitude class of words [20]. That is, given a word, the i-entry of the associated vector was 1 if the word was in the i-th sentence, and 0, otherwise. But, the resultant matrix of word by sentence was very sparse. Thus, in this work we decided to represent an attitude word by means of the vector of words used in the same sentences, and similarly for polarity. This allows us to obtain a more condensed matrix that preserved the same assumptions.

Finally, taking into account the overlap of the classes inherent to the Appraisal Theory, which we also found in polarity as well, but to a lesser extent than in affect, appreciation, and judgment (see next section), we consider that some words may potentially be in more than one of these classes. Therefore, we do not treat either polarity or attitude classification as a multi-classification problem. But, we provide a binary classifier for each polarity and attitude class. For example, for affect, we take as positive examples all word vectors labeled as affect, and as negative examples the remaining vectors labeled as appreciation and judgment.

3 CORPUS STRUCTURE

We assume that words attitude can be determined by the context (the vocabulary) where they tend to appear. We use a Spanish corpus of sentences to study a word context. Since these sentences are in Spanish, we do not use any translation tool to obtain them. Besides of the corpus, we gathered a manually annotated word list in Spanish for each attitude class. The words were classified according to the context where they were actually observed in the corpus. That is, a given word was only manually classified with an attitude if it was observed in any sentence belonging to this attitude type. But, these lists did not have enough words; therefore, we increased them by adding new words used by Turney and Littman, available in the General Inquirer lexicon, which we translated using Power Translator system. We removed words resulting erroneous (i.e. non appraisal word or phrases) and we did not preserve their polarity from English. Thus, all words (previous and new) were annotated considering all its possible uses, without taking into account the corpus. That increased the overlap among the compiled lists (see Table 1). We actually got an overlap of 45.2% for affect, 68.9% for appreciation, 63.6% for judgment, 7.6% for negative, and 10.25 for positive class. We took this lexicon of words manually classified as a reference of "*good-classification*" to compare the results

automatically obtained by Support Vector Machine and Random Forest classifiers.

Table 1. Statistics for previous and current data collections.

List	Previous	Current
Affect	352	672
Judgment	287	1 806
Appreciation	788	1 758
Positive	573	1 268
Negative	389	1 702
Corpus		
No. sentences	1 408	56 970
No. words	32 920	1 358 727

For corpus construction, we manually extracted sentences selected from movie reviews in Spanish, gathered from the website *ciao.es*. Sentences that were considered as containing words expressing some attitude class were selected from each review, these sentences are referred as "attitudinal sentences". This website contains reviews about many items; namely, movies, books, cars, cookware, phones, hotels, music, computers, and others. We decided to select movie reviews given that a great variety of appraisal expressions can be observed.

In addition, we included editorials (or opinion articles). These texts are elaborated by communication specialist, such as journalists and editors. Thus, an editorial shows a more elaborated writing style than that in reviews, allowing other elaborations of appraisal expressions. On the other hand, these texts present a novel topic of public interest, which are usually related to politics, economy, society, art, and sport. That helps to increase the number of judgment expressions. We selected editorials of *excelsior.com.mx*, corresponding to the years 1998, and 1999[3].

Finally, we completed the corpus with sentences that were automatically extracted, containing any word from the translation that we did to increase the lexicon. These new sentences were obtained from online texts of an audible book Hispanic library, with 3200 freely available books. We selected 20 texts of stories and poems. Besides, we added letters from a book collection in Spanish, taken from the Project Gutenberg EBook. Both source are freely available.[4,5] These latter texts

3 http://www.exonline.com.mx/home/.

4 http://leemp3.com/

present a more formal style of writing than movie reviews, also different from the editorials style. The editorials are opinions from a personal and critic author viewpoint, whereas stories, poems and letters usually describe sentiments that exemplify emotional states. Our aim was to increase the affect expressions in the corpus.

Corpora are resources very often used in text processing tasks that are approached with machine learning. The corpus composition can influence the quality of learning and therefore the result might not be as expected. On the other hand, elaborating a customized corpus usually becomes a hard work considering the human effort and time required to achieve it, and even so these efforts could not guarantee a 100% quality. However, we can compute some features of our corpora, which can provide indication that we can achieve the desired results. Thereby, we have used The Watermarking On-line Corpus System (WaCOS) [21]. This tool provides a supervised (corpus and gold standard are required) and unsupervised (only the corpus are required) measure set to evaluate features like domain broadness, shortness, stylometry, class imbalance, and structure. We only used the first two.

The domain broadness of a given corpus is measured by the semantic relation among the categories of the texts that form the corpus. These relationships determine whether the corpus domain is narrow (closely related) or wide (unconnected). In this work, we calculated the domain broadness with the Unsupervised Vocabulary-Based (UVB) measure, which is based on vocabulary dimensionality. Let C be a corpus of n sentences, UVB assumes that if the sentences of C share the maximum number of unique term, then the domain of C is narrow.

$$UVB = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{|V(o_i)| - |(V(C))|}{|C|}\right)^2}\ , \qquad (1)$$

where $V(C)$ is the corpus vocabulary and $V(o_i)$ is the vocabulary of the i-th sentence, and $|C|$, $|V(o_i)|$ and $|(V(C))|$ are the cardinalities of C, $V(o_i)$ and $V(C)$ respectively.

The average vocabulary and text length of this corpus are used to approximate the shortness degree. This is calculated using the Shortness-based measure, VDR.

We work with sentences and we know that they are short texts; we also select texts from different source, i.e. reviews, editorial, stories,

5 http://www.gutenberg.org

poems, and letters. We were also interested in knowing the complexity of the selected sentences, i.e., the rate of the average vocabulary for sentence by average sentence size, Vocabulary vs. Document cardinality Ratios (VDR),

$$VDR = \frac{\log\left(\frac{1}{n}\sum_{i=1}^{n}|V(o_i)|\right)}{\log\left(\frac{1}{n}\sum_{i=1}^{n}|o_i|\right)}, \quad (2)$$

where $V(o_i)$ is the vocabulary of the i-th sentence, and $|V(o_i)|$ and $|o_i|$ are the cardinality of $V(o_i)$ and o_i, respectively.

In addition, we evaluated these measures on two others corpora used in Sentiment Classification task; the SFU Review Corpus[6] of movie, book, and consumer product reviews and Pang & Lee´s[7] corpus of 5000 subjective and 5000 objective processed sentences. The statistical properties for these two corpora are taken as reference to compare our results; and they are displayed in Table 2. In this table, we observe that the three corpora have similar properties, considering domain broadness and shortness, but varying in size.

Table 2. Statistical properties of SFU, Pang & Lee, and our corpus.

Corpus	*UVB*	*VDR*	Total Terms	Corpus Vocabulary Size
SFU	12.19	0.96	95 184	14 801
Pang & Lee	11.70	0.96	231 001	23 926
Ours	21.45	0.95	1 603 234	118 552

When we observed UVB results in Fig. 2, we noted that our corpus has a narrow domain. It show that word senses are constrained to its use in the corpus domain, therefore the automatic classification of some words can lead to results that we do not desire, considering that it was manually classified, estimating many of its possible uses. On the other hand, if the domain is narrow then the vocabulary will be limited, which could complicate the classification of new words.

[6] http://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html
[7] http://www.cs.cornell.edu/people/pabo/movie-review-data/

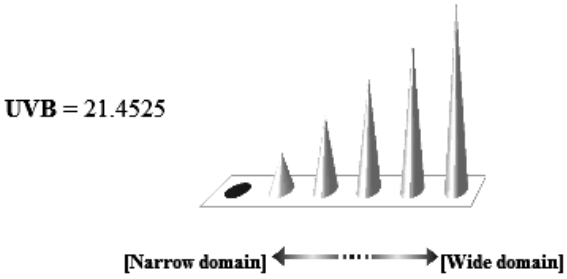


Fig. 2. *Unsupervised Vocabulary-Based* (*UVB*) result for our corpus.

On the other hand, VDR value indicates the average complexity of our corpus, considering that higher complexity implies a bigger vocabulary for each sentence (see Fig. 3.). Therefore, given a word w_i of an attitude categories a, the higher the VDR measure, higher is the probability that w_i occurred in $o \in O$ since we know that $w_{j \neq i} \in a$ occurred in o.

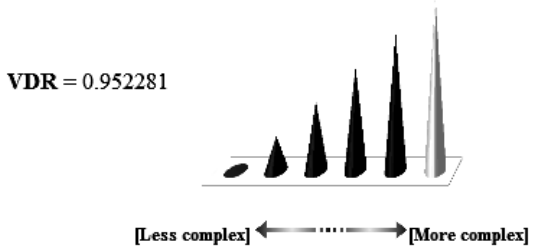


Fig. 3. *Vocabulary vs. Document cardinality Ratios* (*VDR*) result for our corpus.

As mentioned in Section 2, given word to classify, this is represented by a vector of the corpus vocabulary, where the i-entry of the associated vector is 1 if the word occurred with the i-th term of corpus in any sentence, and 0, otherwise. Previously, the sentences were pre-processed obtaining lemmas for each word. To accomplish it, we used the TreeTagger; a system for part-of-speech and extraction of lemmas of the words in a text, developed by Helmut Schmid at the University of Stuttgart[8] [22].

[8] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

4 EVALUATION

We have not found related works with which we can compare our results. Therefore, we use as reference of "*good-classification*", the five lists of words manually classified. Thus, the obtained results are compared against the human judgment. As it was explained in a previous section, we use as positive examples the words annotated in each class, and as negative examples the words in the opposite class, excluding the overlap with positive class. Since the number of examples could be minority in the negative set, we used an over-sampling method called Smote (Synthetic minority over-sampling technique), that increases the proportion of the instances in this set.

Regarding the classifiers, we used two of the suite of Data Mining algorithms that Weka system[9] provides; namely, Support Vector Machine (SVM), and Random Forest (RF). We maintained default parameter values for RF classifier, but for SVM, we opted for the probabilistic version of this algorithm, by setting as true the "buildLogisticModels" parameter, and we used a "PolyKernel" of degree 2. To measure the performance of classification, we divided each set in 50% to train and 50% to test, and computing Precision, Recall, and F-Measure (see Table 3).

The class *appreciation* is more clearly learned with RF than *affect* and *judgment*, in terms of F-measure, possibly because is less ambiguous. Also, RF showed an acceptable F-measure when classifying *positive*. We can note that SVM and FR algorithms show a good performance of attitude classification when we compared them with the human judgment. In all except one of the cases, the Recall is above 50%, with the a*ppreciation* class having lowest value. In terms of the values obtained for Precision, we observed that more than half are above 50% and up to 85%. These preliminary results are encouraging but still vague to make a concluding decision, and could happen because the terms selected to represent the words of attitude do not discriminate the classes appropriately. On the other hand, when we increased the previous words list and annotated them without consider their actual use in the corpus, we probably introduce some noise in the gold standard that we employed as classification references.

[9] http://www.cs.waikato.ac.nz/ml/weka/

Table 3. Precision, Recall, F-Measure of SVM, and RF classifiers for attitude.

Class	Precision	Recall	F-Measure		Precision	Recall	F-Measure
Positive							
SVM	0.64	0.53	0.58	**RF**	0.58	0.68	0.62
Negative							
SVM	0.49	0.77	0.60	**RF**	0.48	0.65	0.55
Affect							
SVM	0.60	0.51	0.52	**RF**	0.51	0.65	0.57
Judgment							
SVM	0.39	0.56	0.46	**RF**	0.42	0.65	0.51
Appreciation							
SVM	0.85	0.42	0.56	**RF**	0.81	0.61	0.70

Sentiment Classification is a difficult task that tries to discover the subjectivity in texts and it is further complicated when we works with noisy unstructured texts which prevail in the actual world. We expect that this performance could be improved if the sentences are reasonably well-separated in its different messages, and we do not consider all words in a sentence to be related with the word that we want to classify. We could only consider the words inside a window for representing the attitude, as well as we could use some part-of-speech tagging or shallow parsing tools to split the sentences.

5 CONCLUSIONS

In this paper, we showed classification of attitude words, which are words that convey the evaluation of sentiments or emotional states, about human behaviors, objects, processes, or people. Beside, these words can express affect, judgment, and appreciation; either positive or negative, according to the Appraisal Theory of language. Thus, we present an improved version of our experimental data collection.

The preliminary results show a good performance of the proposed classification strategy when is compared against human judgments. But, we noted that more than one item sometimes can be evaluated inside a single sentence. This is contrary to our assumptions (that in a single sentence, the evaluation of a single item prevails). Therefore, to reach a final conclusion in future work, we plan to use a window of certain number of words instead of whole sentences, to reduce the terms in the sentences used to represent the attitude words. In addition,

we will work in classification of expressions (word sequences) rather than individual words.

ACKNOWLEDGMENTS. The first author was supported by the scholarship 229536 granted by CONACYT, while second author was partially supported by SNI, México.

REFERENCES

1. Hunston, S., Thompson, G.: Evaluation in text: Authorial stance and the construction of discourse. Oxford: Oxford University Press (2000)
2. Kaplan, N.: Nuevos Desarrollos en el Estudio de la Evaluación en el Lenguaje: La Teoría de la Valoración. Boletín de Lingüística, Universidad Central de Venezuela, pp. 52-78 (2004)
3. Kaplan, N.: La Construcción Discursiva del Evento Conflictivo en las Noticias por Televisión. Ph.D. Thesis, Universidad Central de Venezuela, Facultad de Humanidades y Educacion (2007)
4. Martin, J. R., White, P.R.R.: The Language of Evaluation: Appraisal in English. Palgrave, London (2005)
5. White, P.R.R: Un recorrido por la teoría de la valoración. (Translated by Elsa Ghio). In The Appraisal Website: Homepage. http://www.grammatics.com/appraisal /. [consulted:26/06/2010] (up-to-date 2005)
6. Balahur, A., Montoyo, A: Multilingual Feature–Driven Opinion Extraction and Summarization from Customer Reviews. In Procs. of the 13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems, LNCS, Vol. 5039, pp. 345-346 (2008)
7. Sun, Y. J.: Market and Strategic Analysis of Opinion Aggregators. Master Thesis, Composite Information Systems Laboratory (CISL), Massachusetts Institute of Technology (2008)
8. Wiebe, J., Rapaport, W.: A Computational Theory of Perspective and Reference in Narrative. In Procs. of the Association for Computational Linguistics, pp. 131--138 (1988)
9. Wiebe, J.: Recognizing Subjective Sentences: A Computational Investigation of Narrative Text. PhD Thesis. State Univ. of New York at Buffalo (1990)
10. Whitelaw, C., Navendu, G., Argamon, S.: Using Appraisal Groups for Sentiment Analysis. In Procs. of the 14th ACM International Conference on Information and Knowledge Management, pp. 625--631 (2005)
11. Bloom, K., Garg, N., Argamon, S.: Extracting Appraisal Expressions. In: Proc. of North American Chapter of the Association for Computational Linguistics - Human Language Technologies Conference, pp. 308--315 (2007)

12. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity: an exploration of features for phrase-level sentiment analysis. Computational Linguistics, vol. 35, no. 3, pp. 399--433 (2009)
13. Taboada, M., J. Brooke and M. Stede: Genre-Based Paragraph Classification for Sentiment Analysis. In Procs. of 10th Annual SIGDIAL Conference on Discourse and Dialogue, pp. 62—70 (2009)
14. Banea, C., Mihalcea, R., Wiebe, J., Hassan, S.: Multilingual subjectivity analysis using machine translation. In Procs. of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 127—135 (2008)
15. Bautin, M., Vijayarenu, L., Skiena, S.: International sentiment analysis for News and Blogs. In Procs. of ICWSM 2008, International Conference on Weblogs and Social Media (2008)
16. Brooke, J., M. Tofiloski and M. Taboada: Cross-Linguistic Sentiment Analysis: From English to Spanish. In Procs. of RANLP 2009, Recent Advances in Natural Language Processing, pp. 50--54 (2009).
17. Turney, P.D., Littman, M.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report ERC-1094 (NRC 44929), National Research Council of Canada (2002)
18. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association, ACM Trans. on Information Systems (TOIS), Vol. 21, No. 4, pp. 315--346 (2003)
19. Taboada, M., Grieve, J.: Analyzing appraisal automatically. In Procs. of AAAI Spring Symposium on Exploring Attitude and Affect in Text, pp. 158-161 (2004)
20. Hernandez, L., López-López, A., Medina, J. E.: Recognizing polarity and attitude of words in text. In Procs. of EPIA 2009, 14th Portuguese Conference on Artificial Intelligence, Chapter 11, Text Mining and Applications, pp. 525-536 (2009)
21. Pinto, Avendaño, D.E. : On Clustering and Evaluation of Narrow Domain Short-Text Corpora. PhD Thesis, Universidad Politécnica de Valencia (2008)
22. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In Procs. of International Conference on New Methods in Language Processing, pp. 44-49 (1994)

LARITZA HERNÁNDEZ
INSTITUTO NACIONAL DE ASTROFÍSICA, ÓPTICA Y ELECTRÓNICA, MÉXICO
E-MAIL: < LARITZAHR@INAOEP.MX>

AURELIO LÓPEZ-LOPEZ
INSTITUTO NACIONAL DE ASTROFÍSICA, ÓPTICA Y ELECTRÓNICA, PUEBLA, MÉXICO
E-MAIL: < ALLOPEZ @INAOEP.MX>

JOSÉ E. MEDINA
ADVANCED TECHNOLOGIES APPLICATION CENTER,
LA HABANA, CUBA
E-MAIL: < JMEDINA@CENATAV.CO.CU>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 285–300
RECEIVED 02/11/10 ACCEPTED 21/12/10 FINAL 11/02/11

Features of Latinate Etymologies on the Tasks of Text Categorization

WANYIN LI AND ALEX CHENGYU FANG

City University of Hong Kong, Hong Kong

ABSTRACT

In European languages development, new words and/or new terms are mostly formed using Latin and/or Greek word elements. Linguists have proved the importance of these elements in forming an important part of LSP word stock in different subject fields such as computer science and medicine. This article attempts to show the effectiveness of using words with Latinate etymologies as features for the tasks of text categorization (TC). One essential step for training classifiers in TC to be more accurate is the effective feature selection. Lots of methods for feature selection have been discussed by computer scientists on the basis of information theory and from the perspective of statistics. To cope with the bottleneck of high dimensionality from bags of words (BOW), the core features to be discussed in this article are selected according to etymological information of words and therefore are drastically different from existing feature selection methodologies. The result is analyzed from evaluation schemes including accuracy, precision, recall, and F-measure. The experiment shows that the Latinate words as discriminative features are effective to reduce the dimensionality of the feature space and outperform other feature combinations in F-value of up to 98% when using one of the state-of-the-art methods, i.e., Support Vector Machine from WEKA.

KEY WORDS: *Text Categorization, Feature Selection, Etymology, Latinate Token, Lexicon, Classifier, BOW, Naïve Bayes, Decision Tree, SVM.*

1 INTRODUCTION

With the explosive growth of the Internet, more research efforts are required for the task of text categorization to improve accuracy and efficiency. Two key steps involved in text categorization include the selection of features and the training of classifiers. Reports on feature selection methods with the purpose of scaling down the feature space are usually based on statistical theory and machine learning such as information gain [7, 8, 12, 25], mutual information [13, 25], Chi-square [8, 12, 25] etc. It has been stated [21] that most of the dimensions from bags of words (BOW) are not typically relevant to text categorization and even introduce noise features even though they are said to be statistically important, hence eventually hurt the performance of the training classifiers. Actually, even the proved most successful training classifier for the tasks of text categorization like Support Vector Machine (SVM) is inefficient when directly taking the words/phrases based on the statistical theory as features [27]. It is thus necessary to select distinguishing features according to a methodology different from the conventional statistically based machine learning methods.

This article, taking a different direction from the previous research studies, proposes the selection of core features according to words of a Latin origin based on a lexicon that contains etymological information. Six feature sets are set up including BOW, Latinate words, and the content words from nouns, verbs, adjectives and adverbs. Three selected classifiers, namely SVM, Bayes, and decision tree, are then applied on the above feature sets to find their supportive votes to each of the feature sets. The experiments consistently indicate that the feature set of Latinate words outperforms the other feature sets based on the three selected classifiers.

The rest of this article is organized as follows. Section 2 investigates the rationale behind the use of Latinate words as discriminative features. Section 3 reviews the related studies. Section 4 describes the approach on BNC written texts categorization. Section 5 discusses the result of the experiment and Section 6 concludes the article.

2 WHY LATINATE WORDS

Two lexicographical resources are used in this study. One is a 100 million word collection named British National Corpus (BNC) with

90% of its content coming from written texts. The written part comes from regional and national newspapers, specialist periodicals and journals, academic books and popular fiction, published and unpublished letters and memoranda, school and university essays. The version used in this paper contains texts primarily in 1990's. Another resource contains total 249,331 entries, named Collins English Dictionary (CED), which is a collection of total 128 different languages of etymological knowledge including Latin, French, and Greek etc for contemporary English. The etymological resources in CED are identified by three different tags which output total of 48,593 entries as the final etymological origins.

Recall in the tasks of TC, features drawn from top ranked list based on statistical feature selection face problem that the selection algorithms are either over-relied or mislead by infrequent terms. The way is to select a small number of such feature straightly which may be sufficiently efficient and preserves the relevant information to the texts without utilizing statistics based feature selection. Similar idea is supported by [9] Forman that "additional complexity of feature selection can be omitted for many researchers who are not interested in feature selection, but simply need a fixed and easily replicable input representation".

In summary, the intuition behind the use of Latinate words as features is based on three observations. The first observation is that the linguists have proved that over 70% of the words used in modern English have been borrowing extensively from other languages, especially from Latin, French and Greek [10, 11, 23]. The second observation is that certain contexts from special domains (such as the domain of medicine) would be more likely to turn to certain type of foreign words (such as Latinate words) [4, 24]. The third observation based on a previous work [5] proves that even the density of Latinate words from the distinct text domains performs acceptable to distinguish the spoken and written BNC texts in an average precision rate of 80%. This work aims to further disclose whether the words having Latinate etymologies are effective in distinguishing BNC written texts.

3 PREVIOUS STUDIES

Feature selection is the first key step in a successful task of text categorization. The previous works for this step were mainly based on machine learning theory to pick up the features which are significant in

statistical calculation. When reviewing previous studies with respect to this step, we have in mind two motivations: (1) what indexing terms were selected as features and (2) what weighting schemes were applied to score the selected terms. The features have been typically discussed in relevant studies including single tokens/words [12], keywords [1, 25], bi-grams/n-grams [17, 20], noun phrases [3, 26], and syntactically bounded phrases [16, 18] with respect to the predefined/learned syntactic patterns. Among these features, it has been reported that "stemmed or un-stemmed single words as features give better classifier performance compared with other types of features" [3]. With the summarization on these reviewing facts, this paper makes the typical choice of the indexing terms as different individual tokens like BOW, Latinate tokens, and stemmed tokens from four types of content words.

For weighting schemes, studies [8, 22] have reported that traditional feature selection methods may be over relied when identifying statistical significant features. Likewise, Olsson [12] reported that χ^2 are known to be misled by infrequent terms. Liao [3] also concluded that LOG(tf).IDF as feature weight gives better classifier performance than other types of feature weighting schemes. In this work, we utilize the interface of StringtoWordVector provided by WEKA to transform the selected features into tf*idf vectors as the input of learning algorithms supported by WEKA as well.

To the end of learning algorithms on the tasks of TC, the theory from machine learning is usually applied, like Naïve Bayes Classification [6, 25], Support Vector Machine [6, 8, 15, 25], KNN [6, 8, 12], Decision rule/tree [2, 19] etc. Damerau and Weiss stated in [2] that machine generated decision rules able to compete with human performance in text categorization, whereas Joachims proved that better result can be achieved by using Support Vector Machines [14]. J. Wulandini [6] also showed SVM performs the best compared with Naive Bayes and KNN with 92.5% of accuracy. In this work, we select Naive Bayes, C4.5 Decision Tree, and SVM from WEKA as the learning algorithms to evaluate the candidate feature sets.

4 THE PROPOSED METHOD

The architecture of machine learning based text categorization consists of two main parts (1) Feature selection procedure to identify candidate features; then reduce the feature space using filters; and finally transform the selected textual features into numerical values. (2) The

learning procedure to learn a training model and classify the testing data. Fig. 1 gives the full picture to describe the standard text categorization procedure. This work focuses on the first part, hence, we describe the procedures 1.1, 1.2 and 1.3 which while fulfilled will append the candidate features into the scalable features database discussed in the next section.

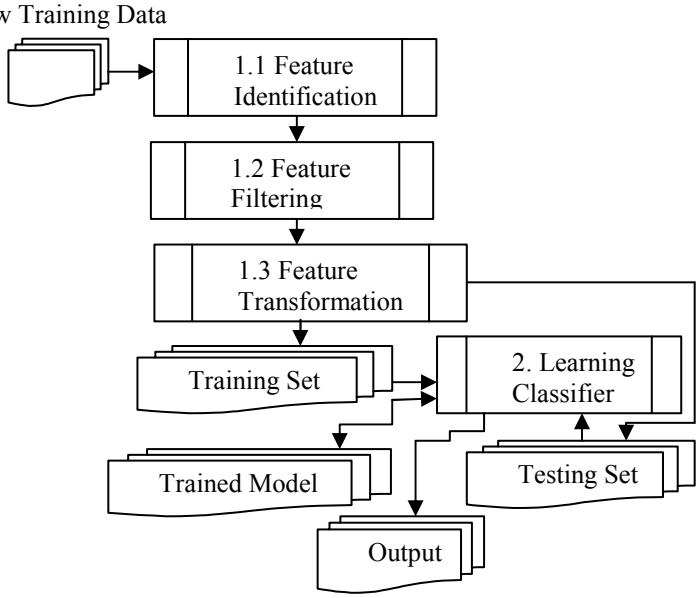


Fig. 1. Architecture of ML-based TC system

4.1 BUILDING UP THE FEATURES DATABASE

In order to compare with the other types of features, we include BOW, Noun/Verb/Adjective/Adverb tokens as well. Fig. 2 is the framework to identify the different types of candidate features mentioned in the process 1.1 in Fig. 1.

BNC has been well annotated in terms of tokenization, lemmatization as well as part-of-speech (POS) tags. Upon the feature identification procedure, in process 2.1, all headwords (HWs) with the tag of POS are directly extracted. As BNC corpus has been annotated

by stemmed tokens for each of the HWs, process 2.2 employs a top-2000 frequent wordlists which is calculated based on the whole corpus to filter out the most frequent tokens. The filtered HWs are fed with POS tags to extract the features of Noun/Verb/Adjective/Adverb tokens in the process 2.5.

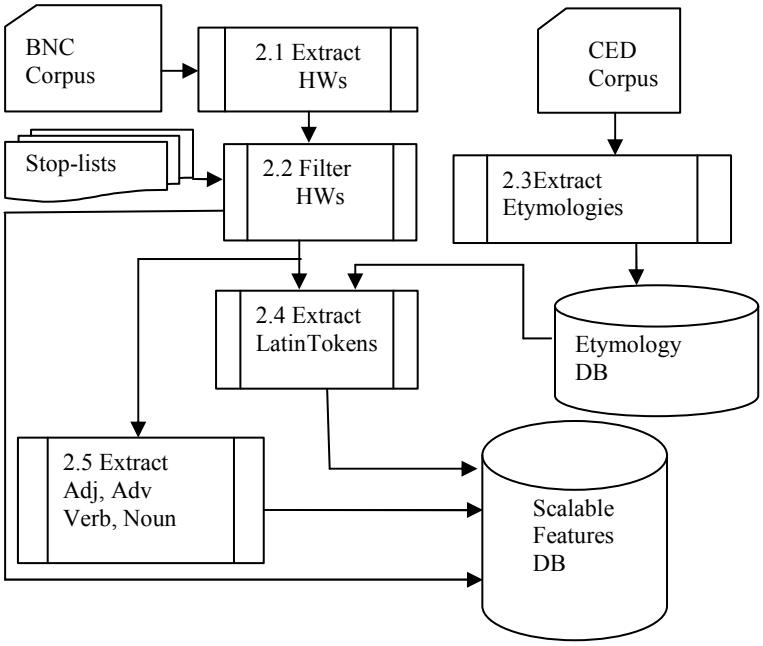


Fig. 2. Framework to build up the features DB

Coming back to the extraction of Latinate tokens, in process 2.3, CED dictionary is firstly utilized to look for root words with etymological origins. These three types are exemplified as follows:

1. cabezon ety Spanish, from cabeza head, ultimately from Latin caput
2. agglutinin ety C19: #7 agglutinate
3. cabinet-making sub head cabinet-maker

The first type assigns Latin as the etymology of the word “cabezon”. In the second case, the etymology of “agglutinin” is assigned by looking for the etymology of “agglutinate”. The third type takes the etymology

of "cabinet-maker" as the etymology of "cabinet-making". All of the three types of tags when traced extract a total of 48,593 etymological entries. For this writing, all entries having Latinate etymology will be further selected from the etymology database as the input to extract Latinate tokens in the process 2.4. The output from the processes 2.2, 2.4, and 2.5 will be appended to the feature database as the input for the selected classifiers in the next section.

For the process 1.2 of feature filtering in Fig. 1, we employ the default filter from WEKA to remove the functional words as well as the words with the frequencies less than 3 for the BOW feature sets to reduce the features space.

For the feature transformation in the process 1.3 of Fig. 1, the standard tf*idf supported by WEKA is applied to index the terms.

4.2 Learning Classifier

To show tokens with Latinate etymologies the discriminative features, Naïve Bayes, C4.5 Decision Tree (J48 in WEKA), and SVM (SOM in WEKA) are selected as learning classifiers to evaluate the selected features. All these classification algorithms are trained using WEKA's default parameter setting such as in SMO using polynomial kernel function, the complexity parameter as C = 1.0 and exponent = 1.0.

5 Experimental Design

The experiment is designed with two testing approaches applied. Firstly, the training models with respect to each learning classifiers are built up based on the train-and-test approach. With the aim of keeping the optimization possible, in the train-and-test approach, each category in the initial BNC corpus has been randomly split into 80% of each as training set to build up the classifier, and an exclusive 20% of each as testing set to test the effectiveness of the classifier. The above splitting is repeated in 5 times. Alternatively, the approach of 5-fold cross validation supported by WEKA is also employed to obtain a macro-averaged performance over the different classes. Table 1 shows the number of even-sized texts under each of the written categories with one sample train-test splitting.

Table 1. Number of texts in written categories.

	Fiction	News	Otherpub	Unpub
Train	2,986	2,856	2,946	2,911
Test	811	826	831	761

5.1 FEATURE SETS

Six datasets such as BOW tokens, Latinate tokens, adjective tokens, noun tokens, verb tokens, and adverb tokens are used in the experiment. Table 2 shows the number of instances in different datasets used as input for the three selected classifiers as well as the time cost by the classifiers to build up the training models.

As stated above, there are total 48,593 etymological entries extracted from CED. Taking the entry list as an input to extract the Latinate tokens, total 2,338 instances of Latinate tokens are returned from the four categories. In the experiment, the number of features in the set of Latinate tokens are approximately the order of 10^5, in the other sets equal to the number of instances times 50. We follow the complexity analysis scheme raised in [21] by using a percentage of features when evaluating the performance of the different types of features, because of the absolute size of tokens vary greatly in the above different feature sets.

Table 2. Number of instances in each dataset.

	No. of Instances		Time to Train Model (minutes)		
	Train	Test	Bayes	J48	SMO
Latinate Tokens	2,338	645	.073	0002	0.17
BOW	75,698	15,140	0006	0469	1982
Adj. Tokens	20,873	5,651	1.13	0093	0099
Noun Tokens	51,409	13,603	3.08	0260	0446
Verb Tokens	34,858	9,291	2.10	0178	0286
Adv. Tokens	12,070	3,363	0.63	19.5	0026

5.2 EVALUATION

We wish to compare the impact of different feature sets based on different evaluation scheme such as precision, recall, Break-even-point (BEP), and macro-average F-measure. With respect to the contingency table described in Table 3, the above named evaluation scheme is defined in formula (1)-(4).

Table 3. Contingency table.

		System Yes	Prediction No
Standard Answer	Yes	*TP*	*FN*
	No	*FP*	*TN*

$$precision: P = \frac{TP}{TP + FP} \quad (1)$$

$$recall: R = \frac{TP}{TP + FN} \quad (2)$$

$$BEP: BEP = \frac{P + R}{2} \quad (3)$$

$$F - measure: F = \frac{2PR}{P + R} \quad (4)$$

Macro-average F-measure is the average on F scores of all the classes.

TP: True Positive results; *FN*: False Negative results
FP: False Positive results; *TN*: True Negative results

5.3 EXPERIMENTAL RESULTS

Table 4 shows the Macro-averaged precision (P), recall (R), BEP (B), and F-value (F) over four categories against the above six features with 5-fold cross-validation for the selected classifiers Naïve Bayes (NB), J48, and SMO respectively.

Table 4. Average performance over four categories with respect to the feature sets.

NB	BOW	Latin	Adj.	Noun	Verb	Adv.
P	.654	**.845**	.652	.608	.624	.708
R	.661	**.814**	.671	.633	.624	.686
B	.657	**.830**	.662	.621	.624	.697
F	.657	**.820**	.662	.620	.624	.697
J48	BOW	Latin	Adj.	Noun	Verb	Adv.
P	.693	**.899**	.525	.540	.542	.566
R	.680	**.890**	.530	.542	.552	.575
B	.687	**.895**	.528	.541	.541	.571
F	.686	**.890**	.524	.540	.545	.569
SMO	BOW	Latin	Adj.	Noun	Verb	Adv.
P	.709	**.983**	.675	.625	.670	.636
R	.690	**.980**	.684	.636	.669	.649
B	.700	**.982**	.680	.631	.670	.643
F	.700	**.982**	.679	.630	.670	.642

Fig. 3 and Fig. 4 show the performance for the conducted feature sets with respect to the three selected classifier in macro-precision and macro-F-value.

The consistent agreement is achieved from both Fig. 3 and Fig. 4 that the best performance was achieved with the feature set based on Latinate tokens. The same conclusion also holds true for all the three selected learning algorithms. Especially in SMO, both the precision and macro-F-value vary in the range of 88% to 98% with the increase of the number of features, which outperform other features such as noun tokens which produced the precision of 62.5% and F-value of 63% on average. The results also disclose that the performance with Latinate tokens as features increases when the number of features in the feature set increases, while this characteristic is not significant for the other feature sets used in our experiments. Except for Latinate tokens, no consistent conclusion can be drawn for the other feature sets with respect to the three algorithms. SMO and J48 give better performance for BOW compared with the features of noun/verb/adjective/adverb tokens, while the worst macro-performance was recorded for adjective tokens.

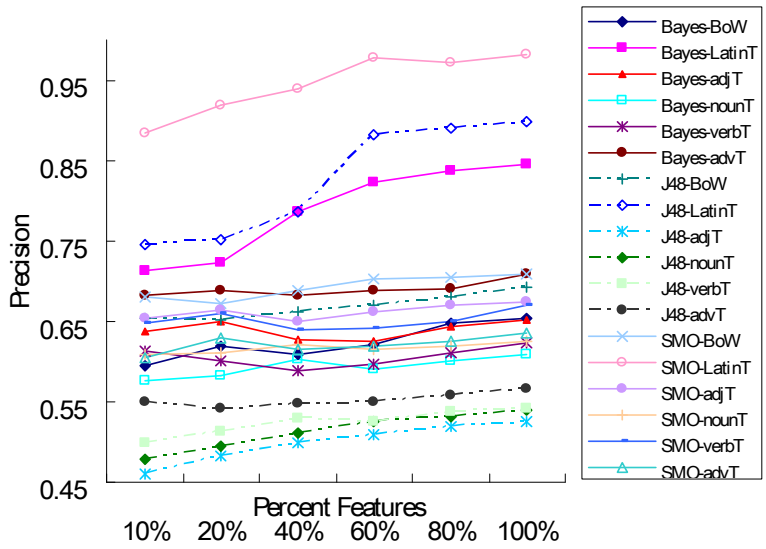


Fig. 3. Precision on six feature sets

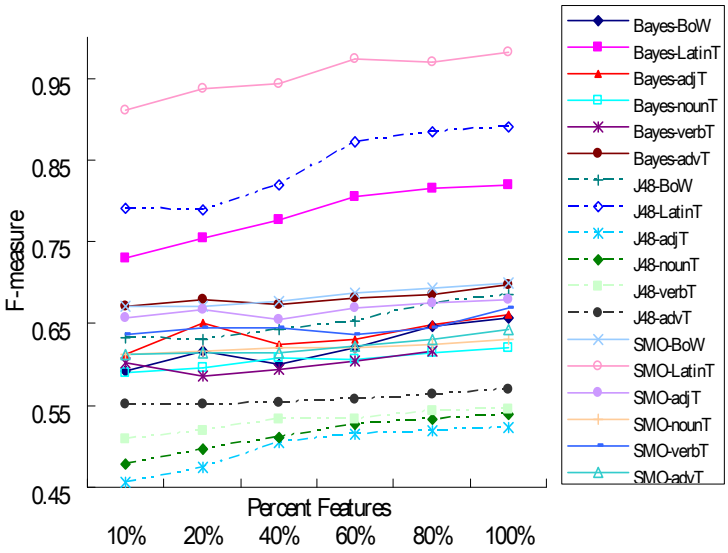


Fig. 4. F-value on six feature sets

6 CONCLUSION AND FURTHER DISCUSSION

This work has tested a number of different feature sets using the learning algorithms of Naïve Bayes, Decision Tree, and SMO from WEKA. The experiments reported in this article concludes that features based on Latinate tokens are superior to the features of BOW as well as stemmed noun/adjective/adverb/verb tokens, in both accuracy and training efficiency. This study finds almost the best performance in the term of F-measure over 98% with the features of the stemmed Latinate tokens reported so far. The result additionally confirms the previous findings [6, 14] that SVM classifier achieves an outstanding performance on the tasks of TC.

Table 4 in Section 5.3 shows that the Latinate tokens as features outperform other selected feature sets. When we look back to try to seek any explanation for the above special result, two reasons appear to be noteworthy for further study. The first reason is that the number of instances is inconsistent in the named feature sets. The number of instances is in terms of thousands in the feature set of Latinate tokens, while it is in terms of tens of thousands in the other four feature sets (Adj/Noun/Verb/Adv). For this observation, we make the number of instance in each of the above four feature sets evenly by randomly selecting thousands from each of them. The second reason is that the number of categories tested in the experiments is small (four in total). To this end, we extend the experiments into eight categories including ACPROSE (academic prose), CONVRSN (conversation), FICTION (fiction), NEWS (news), OTHERPUB (other publication), NONAC (non-academic propose), OTHERSP (other speech), UNPUB (unpublished writing). The results shown in Table 5 agree with the results from Table 4 and in both cases Latinate tokens perform the best among the five feature sets. This thus reinforces our conclusion that Latinate tokens constitute an effective discriminative feature set for the tasks of text categorization.

ACKNOWLEDGEMENT. The research reported in this article was supported in part by research grants from City University of Hong Kong (Project Nos 7008002, 7008062 and 9610126).

Table 5. Performance over eight categories with even number of instances based on the six feature sets.

NB		Convrsn	Othersp	Acprose	Nonac	Fiction	News	Otherpub	Unpub
Latin	P	.923	.734	.76	.779	.799	.853	.853	.693
	R	.712	.715	.73	.684	.667	.735	.736	.72
	B	.817	.725	.745	.732	.733	.794	.795	.707
	F	.804	.724	.745	.728	.727	.789	.790	.707
Adj	P	.702	.671	.182	.194	.518	.573	.314	.615
	R	.745	.606	.522	.398	.714	.592	.473	.521
	B	.723	.638	.352	.296	.616	.582	.393	.568
	F	.723	.637	.269	.261	.600	.582	.377	.564
Noun	P	.694	.696	.412	.559	.709	.692	.594	.493
	R	.786	.581	.648	.330	.878	.555	.309	.244
	B	.74	.567	.469	.511	.703	.588	.526	.540
	F	.737	.538	.462	.507	.703	.569	.517	.536
Verb	P	.601	.508	.361	.391	.612	.528	.467	.325
	R	.794	.598	.439	.414	.790	.586	.494	.580
	B	.697	.553	.400	.402	.701	.557	.481	.452
	F	.684	.549	.396	.402	.689	.555	.480	.416
Adv	P	.727	.631	.548	.265	.651	.641	.400	.282
	R	.778	.586	.598	.368	.789	.567	.375	.333
	B	.752	.608	.573	.316	.720	.604	.387	.307
	F	.752	.608	.572	.308	.713	.602	.387	.305
J48		Convrsn	Othersp	Acprose	Nonac	Fiction	News	Otherpub	Unpub
Latin	P	.589	.690	.736	.821	.736	.821	.916	.826
	R	.621	.723	.488	.738	.932	.833	.928	.870
	B	.605	.706	.612	.779	.834	.827	.922	.848
	F	.604	.706	.587	.777	.822	.827	.921	.847
Adj	P	.419	.314	.336	.304	.407	.412	.261	.321
	R	.402	.334	.356	.305	.452	.373	.250	.312
	B	.411	.324	.346	.304	.430	.392	.255	.316
	F	.410	.323	.346	.304	.428	.391	.255	.316
Noun	P	.498	.275	.386	.221	.237	.229	.142	.217
	R	.394	.427	.339	.154	.142	.435	.141	.130
	B	.446	.351	.362	.187	.189	.332	.141	.173
	F	.439	.334	.361	.181	.177	.300	.141	.162
Verb	P	.408	.337	.317	.265	.372	.281	.239	.315
	R	.387	.337	.364	.235	.416	.361	.218	.278
	B	.397	.337	.341	.250	.394	.321	.228	.297
	F	.397	.337	.338	.249	.393	.316	.228	.295
Adv	P	.505	.402	.354	.203	.330	.502	.219	.191
	R	.537	.502	.253	.170	.388	.526	.202	.195
	B	.521	.452	.303	.186	.359	.514	.211	.193
	F	.521	.446	.295	.185	.356	.514	.210	.193

SMO		Convrsn	Othersp	Acprose	Nonac	Fiction	News	Otherpub	Unpub
Latin	P	.800	.894	.722	.731	.880	.960	.933	.927
	R	.582	.766	.570	.633	.961	.873	.956	.653
	B	.691	.830	.646	.682	.921	.916	.944	.940
	F	.674	.825	.637	.678	.919	.914	.944	.939
Adj	P	.516	.418	.298	.531	.433	.344	.228	.264
	R	.673	.582	.257	.243	.720	.560	.339	.430
	B	.594	.500	.277	.387	.576	.452	.283	.347
	F	.584	.487	.275	.333	.541	.426	.272	.327
Noun	P	.617	.357	.446	.273	.622	.351	.280	.441
	R	.696	.596	.481	.237	.551	.591	.441	.375
	B	.656	.476	.463	.255	.586	.471	.361	.408
	F	.654	.446	.462	.253	.584	.440	.342	.405
Verb	P	.532	.400	.302	.314	.558	.323	.304	.298
	R	.698	.588	.305	.282	.607	.506	.392	.354
	B	.615	.494	.303	.298	.582	.414	.348	.326
	F	.604	.476	.303	.297	.581	.394	.342	.323
Adv	P	.601	.532	.502	.220	.476	.452	.264	.211
	R	.740	.696	.402	.283	.629	.632	.303	.264
	B	.671	.614	.452	.252	.552	.542	.283	.238
	F	.663	.603	.446	.247	.542	.527	.282	.234

REFERENCES

1. Anette, H.: A study on automatically extracted keywords in text categorization. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL, pp. 537-544 (2006)
2. Apte, C., Damerau, F. & Weiss, S.M.: Automated Learning of Decision Rules for Text Categorization. ACM Transactions on Information Systems, v.12, pp. 233-251 (1994)
3. Liao, C., Alpha, S. & Dixon, P.: Feature Preparation in Text Categorization. In: Proceedings of Australian Workshop of Data Mining in CEC (2003)
4. Cabré, T.M.: Terminology: Theory, Methods, and Applications. Amsterdam; Philadelphia: John Benjamins Pub. Co (1999)
5. Fang, A.C., Li, W.Y. & Ide, N.: Latin Etymologies as Features on BNC Text Categorization. In: Proceedings of 23rd Pacific Asia Conference on Language, Information and Computation, Hong Kong, China (2009)
6. Wulandini, F. & Nugroho, A.S.: Text Classification Using Support Vector Machine for Webmining based Spatio Temporal Analysis of the Spread of Tropical Diseases. In: Proceedings of International Conference on Rural Information & Communication Technology, Bandung Institute of Technology, Indonesia, pp. 189-192, (2009)

7. Lee C. & Geunbae, G.: Information gain and divergence-based feature selection for machine learning-based text categorization. Information Processing and Management: An International Journal, v.42 n.1, pp. 155-165 (2006)
8. Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y. & Wang, Z.: A novel feature selection algorithm for text categorization. Expert Systems with Applications: An International Journal, v.33 n.1, pp. 1-5 (2007)
9. Forman, G.: Feature Selection for Text Classification. Chapman and Hall/CRC Press (2007)
10. Greenbaum, S.: The Oxford English Grammar. Oxford University Press (1996)
11. Hughes, G.: A History of English Words. Blackwell Publishers (2000)
12. Olsson, J.O.S. & Oard, D.W.: Combining feature selectors for text classification. In: Proceedings of the 15th ACM international conference on Information and knowledge management, November 06-11, Arlington, Virginia, USA (2006)
13. Pei, Z., Shi, X., Marchese, M. & Liang, Y.: Text Categorization Method Based on Improved Mutual Information and Characteristic Weights Evaluation Algorithms. In: Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, Vol.4 (2007)
14. Joachims, T.: A Statistical Learning Model of Text Classification for Support Vector Machines. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, pp. 128-136 (2001)
15. Joachims, T.: Training linear SVMs in linear time. In: Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 217–226 (2006)
16. Furnkranz, J., Mitchell, T. & Riloff, E.: A case study in using linguistics phrase in text categorization on the WWW. In: Proceedings of AAAI/ICML Workshop (1998)
17. Kanaris, I. & Stamatatos, E.: Webpage genre identification using variable-length character n-grams. In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, Vol.2, Washington, DC, USA, IEEE Computer Society (2007)
18. Lewis, D.D.: Feature selection and feature extraction for text categorization. In: Proceedings of Speech and Natural Language Workshop (1992)
19. Lewis, D.D. & Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. In: Third Annual Symposium on Document Analysis and Information Retrieval, pp. 81-93 (1994)
20. Mansur, M., UzZaman, N., Khan, M.: Analysis of n-gram based text categorization for Bangla in a newspaper corpus. In: Proceedings of the 9th International Conference on Computer and Information Technology, Dhaka, Bangladesh (2006)

21. Rogati, M. & Yang, Y.: High- performing feature selection for text classification. In: Proceedings of CIKM '02, pp. 659–661. ACM Press (2002)
22. Mukras, R., Wiratunga, N., Lothian, R., Chakraborti, S. & Harper, D.: Information Gain Feature Selection for Ordinal Text Classification using Probability Re-distribution. In: Proceedings of the Textlink workshop at IJCAI-07 (2007)
23. Roberts, A. H.: A Statistical Linguistic Analysis of American English. The Hague: Mouton (1965)
24. Stockwell, R. and Minkova, D.: English Words: History and Structure. Cambridge University Press (2001)
25. Wang, G., Lochovsky, F.H. and Yang, Q.: Feature selection with conditional mutual information maximin in text categorization. In: Proceedings of the 13th ACM International Conference on Information and Knowledge Management, ACM, Washington, DC, USA. pp. 342-349 (2004)
26. Zhang, W., Liu, S., Yu, C., Sun, C., Liu, F. & Meng, W.: Recognition and classification of noun phrases in queries for effective retrieval. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, USA (2006)
27. Zhang D. & Lee W.S.: Extracting key-substring-group features for text classification. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM Press, pp. 474–483 (2006)

WANYIN LI
DEPARTMENT OF CHINESE, TRANSLATION AND LINGUISTICS,
CITY UNIVERSITY OF HONG KONG,
HONG KONG
E-MAIL: <CLAIRELI@CITYU.EDU.HK >

ALEX CHENGYU FANG
DEPARTMENT OF CHINESE, TRANSLATION AND LINGUISTICS,
CITY UNIVERSITY OF HONG KONG,
HONG KONG
E-MAIL: <ACFANG @CITYU.EDU.HK >

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 301–317
RECEIVED 28/10/10 ACCEPTED 02/12/10 FINAL 08/02/11

Aspect-Driven News Summarization

JOSEF STEINBERGER, HRISTO TANEV, MIJAIL KABADJOV, AND RALF STEINBERGER

Joint Research Centre, European Commission, Italy

ABSTRACT

A summary of any event type is only complete if certain information aspects are mentioned. For a court trial, readers will at least want to know who is involved and what the charges and the sentence are. For a natural disaster, they will ask for the disaster type, the victims and other damages. Will a co-occurrence or frequency-based sentence extraction summariser automatically provide the requested information, or would the results be better if an information extraction (IE) system first detected the summary-crucial aspects? To answer this question, we compared the performance of a purely co-occurrence-based method with a system that additionally makes use of targeted IE. As each event type requires different information aspects and not all of them were covered by the existing IE software, we used a tool that learns semantically related terms to cover the remaining aspects. The comprehensive evaluation in the TAC'2010 competition showed that event extraction is indeed beneficial for summarisation performance, and that summary quality is directly related to IE quality. Our integrated system was ranked among the top systems participating at TAC.

1 INTRODUCTION

Our main goal is to produce succinct multilingual summaries within the Europe Media Monitor (EMM)[1] framework. The news collator gathers

[1] `http://emm.jrc.it/overview.html`

around $100,000$ news articles every day from various news sources and continuously groups them, producing topic-homogeneous news clusters for each of a set of $40+$ languages. There are thus many news clusters in various languages, varying in size from two to more than a hundred articles. Multi-document summarization systems can potentially reduce this big bulk of highly redundant news data and obtain one succinct text which summarizes the most important content.

Evaluation of multi-document summarisation is difficult and time-consuming. Teams participating in the summarisation task of the Text Analysis Conferences TAC, organised by the US National Institute of Standards and Technology, benefit from a thorough evaluation of the output of competing systems on a standard test set. While the task in TAC'2009 simply was to produce a concise summary of a cluster of related news articles, TAC'2010 requested that a given list of core information aspects for different event types be addressed in the automatic summaries. This ambitious and challenging requirement is congruous with current, IE-aware trends in the field of summarization [1, 2].

In this paper, we present a novel approach to combining standard extractive summarization techniques with higher-level information extraction in a neat and unified manner. By submitting results produced by both this new approach and the standard technique to the TAC'2010 competition, we received a detailed comparative evaluation of both methods, giving us insight in the relative benefits of either approach.

One successful approach to standard summarization (e.g., yielding scores in the top 10% at previous TACs) builds on the Latent Semantic Analysis (LSA) paradigm. Proponents of this approach to summarization include [3, 4]. Being, by definition, a language-independent approach which is one of the core requirements in our setup, we decided to adopt it as a foundation for building an IE-aware summarizer. Additionally, from the news collator project for which we are building the summarization system, a mature multilingual event extraction (EE) system [5] was made available to us. Coincidentally, it was purpose-built for a very similar domain to that of the TAC corpora and as such, it by definition covered several of the aspects specified in the summarization track of TAC'10. In order to cover some of the remaining aspects of the TAC'10 track, we in addition used a system implementing statistical distributional semantics methods to learn new terms lexically similar to an initial seed of terms [6].

In the remainder of this paper we firstly discuss related work (section 2), then, in section 3, we describe the information extraction tools we

used, followed by the description of our hybrid IE-aware summarization approach in section 4. Next, in section 5 we present a detailed analysis of the results obtained at TAC'10, and finally, we conclude and give pointers to future work.

2 Related Work

There is several related work carried out in the past which tried to exploit the potential of using information extraction in summarization. As a pioneering effort, the SUMMONS system [7], which summarized the results of MUC-4 IE systems in the terrorism domain, was the first to suggest using IE in a summarization system, though no evaluation was carried out. In [8] another system that combined information extraction and summarization was presented. Even though the potential improvement in content coverage when simulating the output of the IE system was demonstrated, using the actual output of the IE system was not good enough. Another attempt to use IE and summarization in a sequential pipeline was proposed in [9]. The system dynamically determined the focus of the article (mainly based on the analysis of entity mentions), which in turn determined the specific information that was extracted. However, the study arrived at inconclusive results. In [2] an approach that used templates conceived from the rhetorical structure of scientific papers was proposed. The templates guided the search for appropriate sentences in the source text. In [1] a new set of features based on low-level, atomic events that describe relationships between important actors in a document or set of documents was presented. Using the event-based features resulted in an improvement in summary quality over using lexical features, but also in avoiding summary redundancy.

3 Information Extraction for Summarization

We describe here information extraction components we used to capture the required summary information. For capturing highly frequent topics in a cluster we use in addition to lexical features (words and bigrams) also person, organization and location entity mentions discovered by our entity recognition and disambiguation tools. For capturing the category-related aspects we used our event extraction system and the tool for automatic learning of semantic classes.

3.1 *Entity Recognition and Disambiguation*

Within the EMM's NewsExplorer project[2] multilingual tools for geotagging [10] and entity disambiguation [11] were developed. We used both systems to extract information about mentions of the entities in the TAC clusters. The extracted features were used as additional features in co-occurrence calculation but also to capture several aspects (places of events and persons involved in investigations).

3.2 *Aspects Identified by NEXUS*

NEXUS is an event extraction system which analyzes news articles reporting on violent events, natural or man-made disasters (see [5] for detailed system description). The system identifies the type of the event (e.g., flooding, explosion, assassination, kidnapping, air attack, etc.), number and description of the victims, as well as descriptions of the perpetrators and the means, used by them. For example for the text "Three people were shot dead and five were injured in a shootout", NEXUS will return an event structure with three slots filled: The *event type* slot will be set to *shooting*; the *dead victims* slot will be set to *three people*; and the *injured* slot will be set to *five*. Event extraction is deployed as a part of the EMM family of applications, described in [12].

NEXUS relies on a mixture of manually created linguistic rules, linear patterns, acquired through machine learning procedures, plus domain knowledge, represented as domain-specific heuristics and taxonomies. For example, one of the linear patterns for detection of *dead victims* is *[PERSON-GROUP] were shot dead* . The *[PERSON-GROUP]* phrases are recognized by a finite-state grammar. Event type detection is done through a combination of keywords, a Naive Bayes statistical classifier and several domain-specific rules.

NEXUS has been used to analyze online news in several languages and showed reasonable levels of accuracy [5].

We found out that some of the aspects, relevant to the summarization task, correspond to the information extracted by NEXUS. In particular, the aspects "What happened", "Perpetrators" and "Who affected" have corresponding slots in the event structures of NEXUS.

In our summarization experiments we ran the event extraction system on each news article from the corpus and we mapped extracted slots to summarization aspects. This was done in the following way: The event

[2] `http://emm.newsexplorer.eu/`

type (e.g., terrorist attack) was mapped to the aspect "What happened"; the slot "Perpetrator" was mapped to the aspect "Perpetrators"; and the values for the aspect "Victims" were obtained as a union of the event slots: "Dead victims", "Injured", "Arrested", "Displaced", "Kidnapped", "Released hostages" and "People, left without homes". At the end, from a fragment like: "three people died and many were injured", the system will extract two values for the aspect "Who affected", namely "three people" and "many".

3.3 *Learning Lexica for Aspect Recognition*

Ontopopulis is a system for automatic learning of semantic classes (see [6] for algorithm overview and evaluation). As an input, it accepts a list of words, which belong to a certain semantic class, e.g. "disasters", then it learns additional words, which belong to the same class. Ontopopulis is a multilingual adaptation of a syntactic approach described earlier in [13]. This approach accepts one or several seed sets of terms, each belonging to a semantic class; then, it finds other terms, which are likely to belong to the same semantic class.

Ontopopulis extracts for each semantic class a list of context features, n-grams which tend to occur with the seed set for this class. Each n-gram has a statistical score assigned to it. At the end, for each semantic class, the system finds other terms, which tend to co-occur with its context features. These terms are considered as candidate terms for the corresponding semantic class. For example, if we want to learn words from the class "natural disaster", we can give to Ontopopulis the following seed set *earthquake, flooding, tsunami*. Then, the system learns terms like *mudslides, landslide, tornado, cyclone, flash floods, fire, wildfires*, etc.

Clearly, the system output needs to be manually cleaned, in order to build an accurate lexicon. Since the terms are ordered by reliability (more reliable terms are at the top), the user can review the list, starting at the top, deciding where to stop on the basis of his/her availability or the quality of the list around the point reached within the list. The unrevised items are discarded. Another possibility is to skip the manual reviewing process and take all the terms up to a certain threshold. This approach, however, cannot guarantee very high accuracy.

We learned 4 lexicons, using Ontopopulis, followed by manual cleaning. Each lexicon was relevant to a specific summary aspect. The four aspects covered by our lexicons are: "Damages", "Countermeasures", "Re-

source", and "Charges". Here we give a short sample from each of the learned lexicons:

1. Damages: damaged, destroyed, badly damaged, extensively damaged, gutted, torched, severely damaged, burnt, burned
2. Countermeasures: operation, rescue operation, rescue, evacuation, treatment, assistance, relief, military operation, police operation, security operation, aid
3. Resource: water, food, species, drinking water, electricity, gas, forests, fuel, natural gas
4. Charges: rape, kidnapping, aggravated, murder, attempted murder, robbery, aggravated assault, theft, armed robbery

The words and multi-word terms from these four lexicons were used to trigger the corresponding summary aspects.

4 Sentence Extraction Based on Co-occurrence and Aspect Information

In this section we describe how the extracted information is combined with lexical features to produce summaries that contain frequently mentioned information (derived from co-occurrence analysis) as well as the required aspects.

4.1 *LSA-based Co-occurrence Information*

Originally proposed by [3] and later improved by [14], this approach first builds a term-by-sentence matrix from the source, then applies Singular Value Decomposition (SVD) and finally uses the resulting matrices to identify and extract the most salient sentences.

The LSA approach to summarization first builds a term-by-sentence matrix from the source, then applies Singular Value Decomposition (SVD) and finally uses the resulting matrices to identify and extract the most salient sentences. SVD finds the latent (orthogonal) dimensions, which in simple terms correspond to the different topics discussed in the source.

More formally, we first build matrix $\mathbf{A}$ where each column represents the weighted term-frequency vector of sentence j in a given set of documents. The weighting scheme we found to work best is using a binary local weight and an entropy-based global weight (for details see [14]). If we generalize the notion of term to entail, in addition to words, also entities we can obtain a semantically enriched representation.

After that step Singular Value Decomposition (SVD) is applied to the above matrix as $\mathbf{A} = \mathbf{USV}^T$, and subsequently matrix $\mathbf{F} = \mathbf{S} \cdot \mathbf{V}^T$ reduced to r dimensions[3] is derived. This matrix that is passed to the sentence selection phase represents the topics of the cluster identified by co-occurring features.

4.2 *Aspect Information*

We use the aspects identified by the information extraction tools to boost the co-occurrence-based scores of the sentences that contain the aspects relevant to the corresponding cluster category. For each article cluster we build an aspect-by-sentence matrix **P** which contains boolean values to store the aspects' presence/absence in sentences. For each cluster category a different set of aspects is applied. This matrix is used in the sentence selection process then.

4.3 *Sentence Selection*

Input to the sentence scoring/selection is formed by matrices **F**, containing information about the most important topics within the cluster, and **P**, containing aspect information.

Sentence selection (see figure 1) starts with measuring the length of sentence vectors in matrix **F**. The length of the vector can be viewed as a measure for importance of that sentence within the top cluster topics. We call it 'co-occurrence sentence score'. For the aspect matrix (**P**) we do the same: measuring the length of sentence vectors. In this case the score corresponds to how many relevant aspects the sentences contain ('aspect-based sentence score'). The two scores are then combined in a way that the aspect-based score works as a booster for the co-occurrence score. The formula for the overall score computation is defined as follows:

$$o_j = |\mathbf{f}_j|(1 + |\mathbf{a}_j|^{bc}). \tag{1}$$

where o_j is the overall score of sentence j, $|\mathbf{f}_j|$ and $|\mathbf{a}_j|$ are its corresponding vectors lengths in matrices **F** and **P**. Coefficient bc can control the impact of aspects on the overall score.

[3] The degree of importance of each 'latent' topic is given by the singular values and the optimal number of latent topics (i.e., dimensions) r can be fine-tuned on training data.

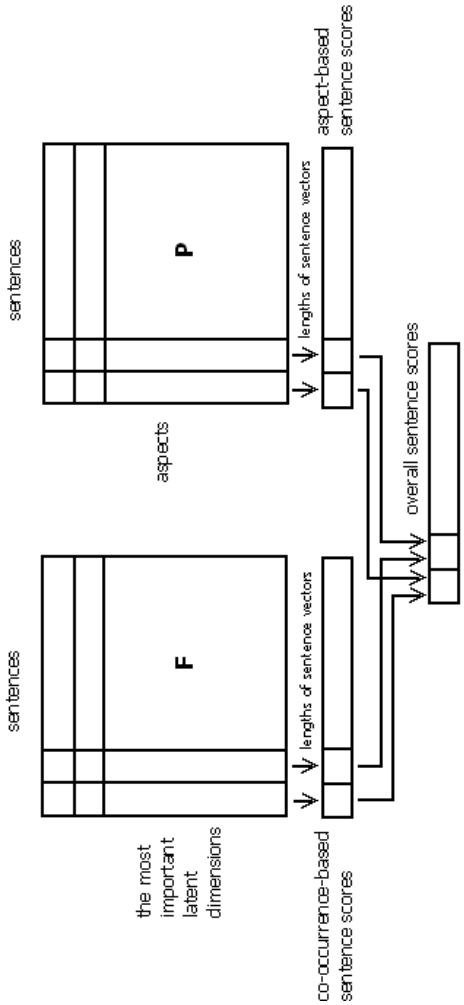


Fig. 1. Sentence selection process.

The sentence with the largest overall score is selected as the first to go in the summary (its corresponding vector in **F** is denoted as $\mathbf{f}_{best}$, similarly $\mathbf{p}_{best}$ for **P**). After placing it in the summary, the topic/sentence distribution in matrix **F** is changed by subtracting the information contained in that sentence:

$$\mathbf{F}^{(it+1)} = \mathbf{F}^{(it)} - \frac{\mathbf{f}_{best} \cdot \mathbf{f}_{best}^{T}}{|\mathbf{f}_{best}|^{2}} \cdot \mathbf{F}^{(it)}, \tag{2}$$

The vector lengths of similar sentences are decreased, thus preventing within-summary redundancy. For aspects, however, we wish to select diverse information as well. But we take a different approach for that. There are cases in which the same aspect should be repeated. For example, for a killing event we want to see the date of the killing and the date when the perpetrator was arrested. Another example are countermeasures. Both following snippets were found important in a model summary of TAC'09 data: *Russian rescue attempts to free and raise the submarine were unsuccessful. Russia requested international help*. Thus, we lower the influence of the aspects already contained in the summary but we do not zero it. Also, we do not use the same formula as in the case of matrix **F** because here we are in positive low-dimensional space in comparison with the positive/negative high-dimensional LSA latent space. We use the following formula to update each value in matrix **P** :

$$\mathbf{p}_{i,j}^{(it+1)} = dc * \mathbf{p}_{i,j}^{(it)}, \quad \text{if} \quad \mathbf{p}_{i,best}^{(it)} > 0. \tag{3}$$

By dc we can control the fadeout of used aspects (a value from 0 to 1).

After the subtraction of information in the selected sentence, the process continues with the sentence which has the largest overall score computed from updated matrices **F** and **P**. The process is iteratively repeated until the required summary length is reached.

5 RESULTS AND DISCUSSION

The task was to produce a 100-word summary for a set of 10 newswire articles for a given topic, where the topic falls into a predefined set of categories. This was similar to last year's task definition (TAC'09), but as oppposed to last year's event, this year's participants (and human summarizers) were given a list of important aspects for each category, and a summary had to cover all those aspects, if possible. The summaries could also contain other information relevant to the topic.

There was an update part of the task this year as at TAC'08 and TAC'09: to produce a 100-word update summary of a subsequent 10 newswire articles for the topic, under the assumption that the user has already read the earlier articles.

The defined categories and their aspects were the following:[4]

1. Accidents and Natural Disasters (1.1 WHAT, 1.2 WHEN, 1.3 WHERE, 1.4 WHY, 1.5 WHO AFFECTED, 1.6 DAMAGES, 1.7 COUNTERMEASURES),
2. Attacks (2.1 WHAT, 2.2 WHEN, 2.3 WHERE, 2.4 PERPETRATORS, 2.5 WHY, 2.6 WHO AFFECTED, 2.7 DAMAGES, 2.8 COUNTERMEASURES),
3. Health and Safety (3.1 WHAT, 3.2 WHO AFFECTED, 3.3 HOW, 3.4 WHY, 3.5 COUNTERMEASURES),
4. Endangered Resources (4.1 WHAT, 4.2 IMPORTANCE, 4.3 THREATS, 4.4 COUNTERMEASURES),
5. Investigations and Trials (5.1 WHO, 5.2 WHO INVOLVED, 5.3 WHY, 5.4 CHARGES, 5.5 PLEAD, 5.6 SENTENCE).

We used several types of information extraction for capturing the aspects. Several aspects were identified by our event extraction system:

- WHAT HAPPENED (used for aspects 1.1, 2.1, 3.1, 5.3): = type of event (e.g. 'bombing');
- WHO AFFECTED (1.5, 2.6, 3.2, 5.1) = number of victims/injured/ displaced etc. (we extracted a full string, not only a number, e.g. '200 soldiers killed');
- PERPETRATORS (2.4, 5.1).

We treated the aspect 5.1 in a special way. For several event types, like 'arrest' the affected person is the one who is investigated, however, for other types of events like 'killing' that person is the perpetrator. This is the reason why we used both WHO AFFECTED and PERPETRATORS slots for capturing the aspect.

The lexical lists of semantically similar terms were generated for capturing the following aspects:

- DAMAGES (1.6, 2.7);
- COUNTERMEASURES (1.7, 2.8, 3.5, 4.4);
- RESOURCE (4.1) = list of resources;
- CHARGES (5.4).

For the identification of temporal expressions (aspects 1.2, 2.2) we produced simple lists of month names etc. Now we work on including a proper temporal analysis.

In the case of aspect 5.2 we took advantage of the fact that we have information about person mentions in the text. This aspect was set in the case that there was a person mentioned in the particular sentence. We

[4] For full definition of the aspects see the official task guidelines: `http://www.nist.gov/tac/2010/Summarization/Guided-Summ.2010.guidelines.html`.

took the same approach for locations (1.3 and 2.3). All locations were considered as fillers of that aspect.

We did not deal with the most complex aspects (1.4, 2.5, 3.3, 3.4, 4.2, 4.3, 5.5, 5.6). We simply rely on the fact that they should be captured by the co-occurrence part of the sentence scorer if they seem to be important (frequently mentioned).

We submitted two runs. The first one (RUN-IE) is the complete proposed system: it combines co-occurrence and aspect information. The second run (RUN-CO) represents our baseline system: it uses only co-occurrence information (including lexical and entity co-occurrence). In the remainder of this discussion we refer to the former run as the IE run and the latter as non-IE run (but note that the non-IE run includes the named entity information).

The summaries were evaluated at the NIST for content (based on Columbia University's Pyramid method [15]), readability / fluency and overall responsiveness. ROUGE [16] and BE [17] scores were also provided.

The total number of systems this year was 43 including two baselines. The 1st baseline (LEAD) was the first 100 words from the most recent document, the 2nd baseline was the output of the MEAD summarizer [18]. 23 groups participated.

We can analyze 3 types of results. The overall results compare the systems based on all 46 topics (clusters) – basic and update summaries. We have also results for each category. But also, we can see how well we identified each aspect (only pyramid scores are available).

5.1 *Overall Results*

Table 1 contains the overall TAC results for initial summaries. We report the results and corresponding ranks (in brackets) within all the 43 systems of the two best TAC systems, our two submissions, and the two baselines.

In the case of initial summaries the run that included aspects (run 25) performed better in the overall responsiveness and linguistic quality than the run based on co-occurrence only (run 31). It was slightly worse when evaluated by the Pyramid method. We do not report here the evaluation of the number of repetitions, but also in this qualitative measure the aspect-based run was better. The reason could be that we try to select diverse aspects here. Overall, both our systems were ranked high in linguistic quality. One reason could be that sentences that contain full entity mentions, which are used as features in the co-occurrence-based

Table 1. TAC'10 results of the Guided summarization task - initial summaries.

Run ID	Overall responsiveness	Linguistic quality	Pyramid score
16 (the best run in Overall resp.)	**3.17** (1)	3.46 (2)	0.40 (4)
22 (the best run in Pyramid score)	3.13 (2)	3.11 (13)	**0.43** (1)
RUN-IE (co-occurrence+aspects)	2.98 (10)	3.35 (4)	0.37 (18)
RUN-CO (co-occurrence only)	2.89 (19)	3.28 (6)	0.38 (13)
2 (baseline - MEAD)	2.50 (27)	2.72 (29)	0.30 (26)
1 (baseline - LEAD)	2.17 (32)	**3.65** (1)	0.23 (32)

part of the sentence scorer, are getting higher scores. They are usually summary-worthy sentences and are less likely to contain anaphoric references to entities in the preceding context. Our systems performed better than both baselines, with the obvious exception of the LEAD baseline and linguistic quality (the summary is formed by a continuous text from one article). The score differences between our systems and the best two listed systems (16 and 22) were not significant[5].

5.2 *Category-focused Results*

Now we continue with the discussion of the results for each category. We report the scores and ranks of both our systems in each cell of the table – the first score and rank correspond to Run 25 (with information extraction-based aspect capturing), the second to Run 31 (co-occurrence only).

Table 2. Scores and ranks of our runs for each category (RUN-IE – RUN-CO).

Category	Overall responsiveness	Linguistic quality	Pyramid score
1. Disasters	3.00 (23) - **3.57 (2)**	3.43 (3) - 3.29 (5)	0.38 (23) - **0.43 (10)**
2. Attacks	**3.71 (3)** - 2.86 (22)	**3.29 (4)** - 3.00 (16)	**0.56 (6)** - 0.49 (18)
3. Health	**2.75 (6)** - 2.42 (21)	3.33 (6) - 3.25 (9)	0.30 (9) - 0.31 (7)
4. Resources	2.50 (25) - 2.60 (21)	3.60 (3) - 3.40 (6)	0.24 (29) - 0.27 (23)
5. Investigations	3.20 (6) - 3.30 (2)	3.10 (10) - **3.40 (2)**	0.45 (14) - 0.47 (5)

[5] Here we omit the discussion of the results on update summarization, since our main interest is in the core summarization task.

In the case of the category "Accidents and natural disasters" the co-occurrence-only approach worked clearly better than the approach with IE. Our simpler run was ranked 2nd in overall responsiveness. The reason of the weaker performance of the IE-based run could be that several times the summarizer selected a sentence that mentioned a historical event, not the event that the cluster was focused on (like a previous earthquake in the same place).

On the contrary, in attacks we can see a really huge improvement with IE: 6th in Pyramids (compared to 18th), 3rd in overall resp. (compared to 22nd) and 4th in linguistic quality (compared to 16th). It could be explained by the fact that this category is the focus of the event extraction system.

In the 'health and safety' category we can notice an improvement when using IE, except for Pyramids. Overall, the runs were ranked high in that category. In the case of 'endangered Resources' the results were poor. We did not focus on this particular category. The linguistic quality, however, showed high levels also for this category.

In the last category, investigations and trials, the system without IE worked better but the differences in the scores were not significant. Our simpler system was ranked high: 2nd in both linguistic quality and overall responsiveness, and 5th in Pyramids.

5.3 *Aspect-focused Results*

In this section we focus on the most fine-grained results: how well each particular aspect was captured. We can use only Pyramid scores for this evaluation. We report the scores and ranks of our systems and the score of the best system. However, the best score refers to a different system for each aspect.

Firstly, we look at the aspects derived from NEXUS (table 3). Clearly, using type of event as capturing the 'what happened' aspect was not successful. An indicator like 'bombing' seems to be too general for capturing what happened. It could be left to LSA to cover this aspect by selecting the most frequent information. In the case of the aspect 'who affected' there was a large improvement for the attacks category. Roughly speaking, there was no effect in other categories. We noticed also an improvement in update summaries for this aspect. The IE run was successful in capturing also the 'perpetrators' aspect in comparison with the run without IE. Compared to other systems, however, the runs were ranked only slightly above the average.

Table 3. Pyramid scores and ranks of our runs for each aspect identified by the event extraction system.

Aspect	RUN-IE (rank)	RUN-CO (rank)	Best
1.1 WHAT (disasters)	0.60 (24)	**0.79 (3)**	0.89
2.1 WHAT (attacks)	0.74 (21)	0.79 (12)	0.88
3.1 WHAT (health)	0.33 (17)	0.36 (14)	0.58
5.3 REASONS (investigations)	0.46 (19)	**0.59 (6)**	0.67
1.5 WHO AFFECTED (disasters)	0.36 (25)	0.41 (23)	0.68
2.6 WHO AFFECTED (attacks)	**0.65 (2)**	0.54 (11)	0.66
3.2 WHO AFFECTED (health)	0.29 (6)	0.31 (4)	0.39
5.1 WHO (investigations)	0.67 (17)	0.65 (19)	0.96
2.4 PERPETRATORS (attacks)	**0.48 (18)**	0.34 (24)	0.69

Next, we look at the aspects derived from the lexical list generated by Ontopopulis (table 4). In the case of damages we can see worse results with IE in the category disasters . Treating all events in the cluster as equal probably led to selecting sentences, and subsequently also damages, concerned with non-central events. In attacks we can observe, that without IE we did not capture any damage (the score is 0), compared to the 4th best performance with IE. 'Countermeasures' was the category where the IE-based run was very successful in all four categories. It suggests the lexical lists were the right choice for treating this aspect. In resource descriptions there was a non-significant improvement with IE. In capturing charges the co-occurrence information itself performed better.

Table 4. Pyramid scores and ranks of our runs for each aspect identified by generated lexical lists.

Aspect	RUN-IE (rank)	RUN-CO (rank)	Best
1.6 DAMAGES (disasters)	0.13 (26)	**0.38 (10)**	1.25
2.7 DAMAGES (attacks)	**0.50 (4)**	0 (30)	0.75
1.7 COUNTERMEASURES (disasters)	**0.34 (7)**	0.19 (29)	0.39
2.8 COUNTERMEASURES (attacks)	**0.34 (18)**	0.20 (32)	0.65
3.5 COUNTERMEASURES (health)	**0.31 (1)**	0.24 (7)	0.31
4.4 COUNTERMEASURES (resources)	**0.36 (5)**	0.29 (12)	0.50
4.1 WHAT (resources)	0.49 (19)	0.46 (25)	0.81
5.4 CHARGES (investigations)	0.33 (27)	**0.47 (11)**	0.72

Among the aspects which were treated by other ways the only successful one was the 'who involved' aspect in investigations. Actually, giving a larger weight to all person mentions did a great job, ranking our IE-based submission as the best one. Treating the place aspect the same way was not successful. For capturing time of the events the co-occurrence-driven approach worked well in the case of attacks (2nd).

There are several complex aspects on which we have not worked yet. However, we find that the co-occurrence analysis is able to capture some of those. For instance, we received top rank for identifying reasons for attacks, but in the 'importance of resource' aspect we did not capture anything.

6 CONCLUSION

We presented an approach to addressing multi-document summarization with an IE-aware perspective. The approach combines a co-occurrence-based summarization system with a mature multilingual event extraction system and a system for the automatic learning of semantically related terms tailored to recognise the required aspects. The results showed positive impact on the clusters that deal with the central focus of the event extraction system - criminal/terrorist attack. Regarding natural disasters, the IE system did not successfully distinguish the recent event from historic events mentioned in the same articles, with a negative impact on summary quality. This can be remedied by preferring information found at the beginning of the articles, or by performing a proper analysis of temporal information in the article. Regarding the coverage of new information aspects, not initially covered by the IE system used, we saw that the automatically generated word lists produced good information extraction and summary results. This shows that we can extend the IE system to more information aspects for which a reasonable base of seed terms can be identified. In the absence of IE patterns to recognise the crucial information aspects, it is more or less left to chance whether these important aspects are covered by the co-occurrence-based summary or not. Our next steps include running and evaluating our IE-aware summarization approach on languages other than English.

REFERENCES

1. Filatova, E., Hatzivassiloglou, V.: Event-based extractive summarization. In: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop

2. Ellouze, M., Hamadou, A.: Relevant information extraction driven with rhetorical schemas to summarize scientific papers. In: Advances in Natural Language Processing. Lecture Notes in Computer Science, Springer Verlag (2002) 629–642
3. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of ACM SIGIR, New Orleans, US (2002)
4. Steinberger, J., Poesio, M., Kabadjov, M.A., Ježek, K.: Two uses of anaphora resolution in summarization. Information Processing and Management **43**(6) (2007) 1663–1680 Special Issue on Text Summarisation (Donna Harman, ed.).
5. Tanev, H., Piskorski, J., Atkinson, M.: Real-time news event extraction for global crisis monitoring. In: Proceedings of 13th International Conference on Applications of Natural Language to Information Systems (NLDB 2008). (2008)
6. Tanev, H., Zavarella, V., Linge, J., Kabadjov, M., Piskorski, J., Atkinson, M., R.Steinberger: Exploiting machine learning techniques to build an event extraction system for portuguese and spanish. Journal Linguamatica: Revista para o Processamento Automatico das Linguas Ibericas (2010)
7. Radev, D., McKeown, K.: Generating natural language summaries from multiple on-line sources. Computational Linguistics **24**(3) (1998)
8. White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., Wagstaff, K.: Multidocument summarization via information extraction. In: Proceedings of HLT. (2001)
9. Kan, M., McKeown, K.: Information extraction and summarization: Domain independence through focus types. Technical Report CUCS-030-99, Columbia University (1999)
10. Pouliquen, B., Kimler, M., Steinberger, R., Ignat, C., Oellinger, T., Blackler, K., Fuart, F., Zaghouani, W., Widiger, A., Forslund, A., Best, C.: Geocoding multilingual texts: Recognition, disambiguation and visualisation. In: Proceedings of LREC 2006
11. Pouliquen, B., Steinberger, R.: Automatic construction of multilingual name dictionaries. In Goutte, C., Cancedda, N., Dymetman, M., Foster, G., eds.: Learning Machine Translation. MIT Press, NIPS series (2009)
12. Steinberger, R., Pouliquen, B., der Goot, E.V.: An introduction to the europe media monitor family of applications. In: Information Access in a Multilingual World Proceedings of the SIGIR. (2009)
13. Tanev, H., Magnini, B.: Weakly supervised approaches for ontology population. In: Proceedings of the 11th conference of the European Chapter of the Association for Computational Linguistics (EACL). (2006)
14. Steinberger, J., Ježek, K.: Update summarization based on novel topic distribution. In: Proceedings of the 9th DocEng ACM Symposium, Munich, Germany. (2009)

15. Nenkova, A., Passonneau, R.: Evaluating content selection in summarization: The pyramid method. In: Proceedings of the Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL). (2004)
16. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out, Spain (2004)
17. Hovy, E., Lin, C., Zhou, L.: Evaluating duc 2005 using basic elements. In: Proceedings of the DUC. (2005)
18. Radev, D., Otterbacher, J., Qi, H., Tam, D.: Mead reducs: Michigan at duc 2003. In: Proceedings of DUC 2003. (2003)

JOSEF STEINBERGER
JOINT RESEARCH CENTRE,
EUROPEAN COMMISSION,
VIA E. FERMI 2749, ISPRA (VA), ITALY
E-MAIL: <JOSEF.STEINBERGER@JRC.EC.EUROPA.EU>

HRISTO TANEV
JOINT RESEARCH CENTRE,
EUROPEAN COMMISSION,
VIA E. FERMI 2749, ISPRA (VA), ITALY
E-MAIL: <HRISTO.TANEV@JRC.EC.EUROPA.EU>

MIJAIL KABADJOV
JOINT RESEARCH CENTRE,
EUROPEAN COMMISSION,
VIA E. FERMI 2749, ISPRA (VA), ITALY
E-MAIL: <MIJAIL.KABADJOV@JRC.EC.EUROPA.EU>

RALF STEINBERGER
JOINT RESEARCH CENTRE,
EUROPEAN COMMISSION,
VIA E. FERMI 2749, ISPRA (VA), ITALY
E-MAIL: <RALF.STEINBERGER@JRC.EC.EUROPA.EU>

IJCLA VOL. 2, NO. 1–2, JAN-DEC 2011, PP. 319–332
RECEIVED 05/11/10 ACCEPTED 26/11/10 FINAL 18/02/11

Translationese Traits in Romanian Newspapers: A Machine Learning Approach

IUSTINA ILISEI[1] AND DIANA INKPEN[2]

[1] *University of Wolverhampton, United Kingdom*
[2] *University of Ottawa, Canada*

ABSTRACT

This paper presents a machine learning approach to the investigation of the translationese effect on Romanian newspapers texts. The aim is to train a learning system to distinguish between translated and non-translated texts. The classifiers achieve an accuracy well above the chance level, the results confirming the existence of translationese manifestation. Also, the experiments investigate whether there are any traits of the simplification universal within translated text. The learning system is enhanced with features previously proposed to stand for this universal, and their impact on the learning model is assessed.

KEYWORDS: *translationese, machine learning, translation studies*

1 INTRODUCTION

Beginning in the eighties, certain studies noted certain characteristics exhibited by translated language compared to non-translated texts. These unnatural 'fingerprints' suspected to be characteristic of translated texts were first described by Gellerstam and the effect was called translationese [1]. In the nineties, the topic was studied intensely and the translation universals theory was proposed by [2, 3] describing four hypotheses: simplification, explicitation, transfer, and convergence. Translated language is believed to manifest certain universal features, as a consequence of the

translation process, translated texts presenting their own specific lexico-grammatical and syntactic characteristics [4–6]. Toury enriched this theory by adopting a different view and by proposing two laws of translation: the law of standardisation and the law of interference [7]. Laviosa continued this line of research by proposing features for the simplification universal in a corpus-based study [8].

However, the existence of translation universals continues to be a polemical and highly debated issue in translation studies. While some scholars, like [9, 10], claim the universality aspect of the proposed hypotheses, others [11] emphasise that the value of these assumptions stands in their explanatory power. Despite some evidence of the existence of such tendencies on translated texts, there is still a remarkable challenge in defining the specific features which characterise each translation universal presented.

On one hand, the main reason to investigate these hypotheses is to raise awareness among translators about their conscious or unconscious effects on translated texts, and the relationship between language and culture [8]. Bringing unconscious tendencies to light will emphasise translators' decisions, and hence should pave the way for future development of more accurate and natural translations, with more "desired effects and fewer unwanted ones" [12].

On the other hand, the automatic identification of these hypotheses' effects on texts can be a module in various natural language processing frameworks: it can improve web-based parallel corpus extractors' by finding the candidate parallel texts [13], or it can be integrated into statistical machine translation systems to learn the direction of the translation [14].

The objective of the current study is to model a language-independent learning system able to distinguish between translated and non-translated texts. The advantages of such a methodology are obvious: the system has a wide applicability for other languages, and thus, the "universal"character of this hypotheses is easier to investigate. An additional goal is to investigate whether the simplification features previously proposed [8, 15], according to the simplification universal, are influencing the learning model. In the same line of research similar supervised learning techniques were employed on different languages, for medical and technical texts in Spanish [16], and also for Italian [17]. It must be noted that most of the studies employed on translationese use a corpus-based approach [3, 8], whereas others adopt the advantages offered by machine learning techniques.

2 RELATED WORK

Translationese has been previously studied by different scholars, most of them employing a corpus-based approach and comparing different patterns between translated and non-translated texts. Interesting patterns have been outlined, with important differences being discovered [18]. However, the inception of the pattern under investigation is almost always based on scholars' intuition, and clearly, the approach adopted is hand-engineered. Machine learning frameworks brought interesting results for translationese [17, 16] and new pathways of research were recently created. These methodologies are obviously complementary and new pathways to further investigations are outlined.

One of the hypotheses of translation theory is the simplification universal. The universal is described as the tendency of translators to produce simpler and easier-to-follow texts [2], and some empirical results sustaining the universal were provided [8]. Laviosa investigates lexical patterns for English and the obtained results show a relatively low proportion of lexical words over function words in translated texts, and a high proportion of high-frequency words compared to the low-frequency words. Moreover, great repetition of the most frequent words and less variety in the most frequently-used words has been emphasised [19].

Furthermore, a corpus-based approach which tests the statistical significance of the features proposed for the investigation of the simplification universal has been presented for Spanish [15, 20]. The experiments were on both the medical and the technical domains, and the translated texts were produced by both professional and semi-professional translators. In [15], the simplification universal is confirmed only for the lexical richness attribute. The results for the following features appear to be against this universal: complex sentences, sentence length, depth of syntactical trees, information load, number of senses per word. The experiments in [20] revealed that translated texts exhibit lower lexical density and richness, seem to be more readable, have a smaller proportion of simple sentences and appear to be significantly shorter, and that discourse markers were used significantly less often. Simplification fingerprints were found on the technical translations and seemed to show that texts written by non-professional translators do not have such simplification traits.

For Italian, a different perspective over translationese is given by the supervised learning approach employed by Baroni and Bernardini [17]. They investigate whether a computer system can distinguish between

translated texts from non-translated ones in the Italian language. A special corpus for this purpose was compiled and the results of an SVM classifier depend heavily on lexical cues, on the distribution of n-grams of function words and on morpho-syntactic categories. In particular, they notice that elements such as personal pronouns and adverbs also influence the framework. Therefore, it is proved that shallow data representations can be sufficient to automatically distinguish professional translations from non-translated texts with a high accuracy, and it was hypothesised that this representation captures the distinguishing features of translationese. Moreover, the difficulty for humans to differentiate translated and non-translated texts is emphasised and explicit evidence of the superiority of automatic knowledge-poor system on the same task is shown. In contrast to their study, current experiments avoid the exploitation of n-gram indicators or any type of language-dependent attributes, being able to reuse the same learning model on various languages. The bag-of-words model (unigrams) was avoided to prevent learning to classify according to texts' topic. Additionally, the Romanian language has not been previously studied from this point of view, and since this effect on translated texts is claimed to be a 'universal', applying the learning model to a new language is a novel contribution in the field.

3 METHODOLOGY

The chosen methodology consists in supervised machine learning techniques, with the aim to model a learning system to distinguish between translated and non-translated texts. Therefore, a training and a test dataset were created, comprising instances from both classes at a ratio of 2:1 of non-translated:translated texts. By using the Weka tool[3] [21, 22], classifiers are trained by including and excluding the attributes proposed for the simplification universal within the feature vectors. As a result, the success rate would indicate whether the model is influenced to some extent by the simplification features.

Probably the best resource for the investigation of translationese is a comparable corpus (containing translated and non-translated texts in the same language) [23], and hence, this approach would avoid any foreign interference [24]. As no study of the Romanian language has been employed for translationese, a dedicated type of resource did not exist. For this reason a comparable corpus has been specially compiled for this

[3] http://www.cs.waikato.ac.nz/ml/weka

task, consisting of newspaper articles published between 2005-2009. The translated subcorpus is collected from the Southeast European Times[4] comprising 223 articles written after the year 2005. The non-translated subcorpus comprises 416 documents from the same time-span, in the same domain, from a reputable newspaper from Romania called 'Ziua'[5]. The corpus has in total 341320 tokens (200211 for the translated subcorpus and 141109 tokens for the non-translated subcorpus). The selected articles are written by various translators, so the possibility of a specific style playing a role in the classification task is avoided. Also, the texts are translated from various languages into Romanian, an advantage that assures a high likelihood that all the discovered patterns are not due to one particular source language.

The collected dataset was randomly divided into a training set of 639 texts and a test set of 148 texts, while keeping the same ratio of translated and non-translated class instances in the training and test set. In order to extract the feature vector for the learning process, all the texts of the corpora were first tagged using the part of speech tagger provided as a web service by the Research Institute for Artificial Intelligence[6], the Romanian Academy [25, 26].

The learning system exploits thirty-eight language-independent features extracted from the tagger's output, including both the 'translationese features' and the 'simplification features'. As the translationese effect is considered to happen at the morphological level of the texts [8, 7], the first set of attributes captures general language features, in the current study being referenced as 'translationese features':

- the proportion in texts of grammatical words (the parts of speech considered to belong to this class: determiners, articles, prepositions, auxiliary verbs, pronouns, conjunctions, and interjections);
- the proportion of nouns in texts;
- the proportion of verbs in texts;
- the proportion of adjectives in texts;
- the proportion of adverbs in texts;
- the proportion of numerals in texts;
- the proportion of pronouns in texts;
- the proportion of prepositions in texts;
- the proportion of determiners in texts;

[4] http://www.setimes.com
[5] http://www.ziuaveche.ro
[6] http://www.racai.ro/webservices/

- the proportion of articles in texts;
- the proportion of conjunctions in texts;
- the proportion in texts of grammatical words per lexical words (the lexical words class is represented by nouns, verbs, adjectives, adverbs, and numerals);
- the proportion of interjections in texts;
- the proportion of proper nouns in texts;
- the proportion of common nouns in texts;
- the proportion in texts of verbs in the first person plural;
- the proportion in texts of verbs in the first person singular;
- the proportion in texts of verbs in the second person plural;
- the proportion in texts of verbs in the second person singular;
- the proportion in texts of verbs in the third person plural;
- the proportion in texts of verbs in the third person singular;
- the proportion of auxiliary verbs in texts;
- the proportion of copulative verbs in texts;
- the proportion of modal verbs in texts;
- the proportion in texts of verbs in the indicative mood;
- the proportion in texts of verbs in the subjunctive mood;
- the proportion in texts of verbs in the imperative mood;
- the proportion in texts of verbs in the infinitive mood;
- the proportion in texts of verbs in the gerund mood;
- the proportion in texts of verbs in the participle mood;
- the proportion of comparative adjectives in texts;
- the proportion of positive adjectives in texts;
- the proportion of superlative adjectives in texts.

The data representation for the learning system comprises all the above parameters and also includes the following previously proposed simplification features [19, 15, 16]:

- the lexical richness as the proportion of type lemmas per tokens;
- the sentence length: proportion of number of words per sentence;
- the word length: characters number normalised by tokens number;
- the number of simple sentences[7] normalised by the number of sentences;
- the information load as the proportion of lexical words to tokens.

[7] Given that the tagger does not provide any syntactic information, the following algorithm has been employed to compute this feature: sentences with one or zero personal verbs are considered to be 'simple sentences'.

Morpho-syntactic categories have been previously used as features in a similar classification task, [17], showing that non-clitic personal pronouns and adverbs are distinguishing features of translationese in a study on Italian texts. Also [16, 27] use similar features, such as proportion of numerals, adjectives, finite verbs, pronouns and nouns are among the most useful attributes in a classification task on Spanish medical and technical texts. The current experiments have the advantage of considering even more in depth each feature, in order to investigate if the subcategories of these morphological features have a particular influence on the current learning model.

The classifiers applied for the categorisation task are the following: Jrip, Decision Tree, Naive Bayes, and SVM [22]. The evaluation results are outlined in the next section. These particular classification algorithms were chosen because the rules produced by Jrip and decision trees classifiers provide an output with what has been learnt, Naive Bayes because it is known to work well with text, and SVM because it is known to achieve high performance.

4 EVALUATION

The results obtained with the data representation including the simplification features are compared to the accuracy obtained by the system without these. The accuracies for the 10-fold cross-validation evaluation on the training data and the accuracy for the test dataset evaluation are reported in Table 1. The training dataset comprises 639 instances (223 for the translation class and 416 for non-translation class instances), and the test dataset selected comprises 148 instances (49 for the translation class and 99 for non-translation class).

Table 1. Classification Results: Accuracies for several classifiers

	Excluding Simplification Attributes		**Including Simplification Attributes**	
Classifier	*10-fold cross-validation*	*Test set*	*10-fold cross-validation*	*Test set*
Baseline	65.10%	66.89%	65.10%	66.89%
Naive Bayes	91.71%	91.89%	95.46%	94.59%
SVM	97.18%	95.95%	98.90%	98.65%
Jrip	92.80%	93.24%	92.80%	97.30%
Decision Trees	92.64%	91.89%	94.52%	95.27%

The baseline classifier considers the majority class from the dataset, therefore the baseline is 64.5% because the dominant class is the non-translated one. The results shown are definitely an argument towards the existence of translationese, an effect that was hypothesised only twenty years ago. The best accuracy is obtained by the SVM classifier with a 98.90% value for the 10-fold cross validation and with a 98.65% value for the randomly selected test dataset. Moreover, the SVM classifier performed very well for the Spanish [16] and the Italian [17] language on the same categorisation task, even though different domains were involved in those experiments. These success rates are impressive and prove, without doubt, that an automatic system is able to distinguish between translated and non-translated texts. However, the removal of the 'simplification features' leads to a slightly decreased accuracy for all the classifiers, the Naive Bayes technique registering the biggest difference of approximately 4%. Nevertheless, as an overall perspective, all the classifiers' performances are outstanding, with accuracies ranging between 91.71% and 98.90%

In order to observe the rules considered by the classifiers, the pruned tree output from the JRip classifier and the Decision Tree output are outlined in figures 1 and 2. These two classifiers provide an intuitive output for more detailed data analysis [28].

```
Rule 1: (LexicalRichness <= 0.492095) and (Prepositions >= 0.106925)
            => class=translated (175.0/9.0)
Rule 2: (Nouns <= 0.302041) and (Prepositions >= 0.089489)
            and (InformationLoad <= 0.001211)
            => class=translated (37.0/0.0)
Rule 3: (Prepositions >= 0.118367) and (InformationLoad <= 0.001507)
            and (SentenceLenght <= 27.333334)
            => class=translated (9.0/0.0)
Rule 4: (CommonNouns <= 0.24838) and (InformationLoad <= 0.001329)
            and (SimpleSentences >= 0.73913)
            => class=translated (8.0/1.0)
Rule 5: (LexicalRichness <= 0.485342) and (Adjectives >= 0.098787)
            and (CommonNouns <= 0.259965)
            => class=translated (4.0/0.0)
Rule 6: => class=non-translated (406.0/0.0)
```

Fig. 1. JRip classifier rules output.

As can be seen in the JRip classifier's output, the first rule, quite frequently used, considers the lexical richness and the proportion of prepositions features, whilst more information for this classifier is provided by

information load, sentence length and the proportion of nouns (the second and third rule). The fourth and fifth rule also use the proportion in texts of common nouns, simple sentences and adjectives. On the other hand, the decision trees classifier seems to give a much higher priority to the proportion of nouns, positioning this feature on the first level of the classification tree (figure 2), while lexical richness does not appear at all among the attributes considered by this classifier. Similar to the JRip output, information load and prepositions have an important role in the categorisation task. Moreover, the decision tree algorithm also considers the common nouns, the word length, the third person singular and first person verbs, the determiners and the adverbs in their last levels of the pruned tree.

```
Nouns <= 0.318261
|  InformationLoad <= 0.001387
|  |  Prepositions <= 0.093886
|  |  |  CommonNouns <= 0.232852
|  |  |  |  VerbsPersOneSingular <= 0.000472: non-translated (4.0)
|  |  |  |  VerbsPersOneSingular > 0.000472: translated (3.0)
|  |  |  CommonNouns > 0.232852: non-translated (29.0)
|  |  Prepositions > 0.093886
|  |  |  VerbsPersThreeSingular <= 0.033898
|  |  |  |  Pronouns <= 0.083582
|  |  |  |  |  Nouns <= 0.311526: translated (171.0)
|  |  |  |  |  Nouns > 0.311526
|  |  |  |  |  |  Adverbs <= 0.045584: translated (14.0)
|  |  |  |  |  |  Adverbs > 0.045584: non-translated (4.0/1.0)
|  |  |  |  Pronouns > 0.083582
|  |  |  |  |  VerbsPersOnePlural <= 0.004237: translated (12.0/1.0)
|  |  |  |  |  VerbsPersOnePlural > 0.004237: non-translated (3.0)
|  |  |  VerbsPersThreeSingular > 0.033898
|  |  |  |  Determiners <= 0.030864: translated (5.0)
|  |  |  |  Determiners > 0.030864: non-translated (6.0/1.0)
|  InformationLoad > 0.001387: non-translated (58.0)
Nouns > 0.318261
|  Prepositions <= 0.12724
|  |  Nouns <= 0.327212
|  |  |  WordLength <= 5.537314: non-translated (36.0/1.0)
|  |  |  WordLength > 5.537314: translated (3.0)
|  |  Nouns > 0.327212: non-translated (269.0/1.0)
|  Prepositions > 0.12724
|  |  InformationLoad <= 0.001662: translated (13.0/1.0)
|  |  InformationLoad > 0.001662: non-translated (9.0)
```

Fig. 2. Pruned tree output from the Decision Tree classifier.

Furthermore, the feature selection evaluators' output is exploited in order to see the ranking of the attributes, regardless of any classifier. The

Information Gain and Chi-squared algorithms provide the information from Table 2. The first twenty-six attributes are shown in the figure, as the rest of them are given a null value, and, consequently, they have been omitted from the table. The ranking provided by these two algorithms gives approximately the same type of information. This tendency is similar to the study on the Spanish language [16].

The first four features which most influence the classification are: information load, proportion of nouns, proportion of prepositions, and lexical richness, two of which are considered to stand for the simplification universal. They are shortly followed by another set of five features: proportion of common nouns, proportion in texts of grammatical words per lexical words, third singular verbs, numerals, grammatical words, and simple sentences. The scores provided for the two ranking filters drop for the rest of the items.

Regarding the simplification features considered in these experiments, the ranking algorithms place three of them among the top most influencing features: information load - actually being the most useful feature of all, lexical richness as has been previously hypothesised [3], and proportion of simple sentences in texts (item ranked among the first also in the study on Spanish texts [16]).

4.1 *The Simplification Learning Model*

In order to bring light on the simplification hypothesis, the learning model has been trained using only the simplification features (information load, lexical richness, sentence length, word length, and simple sentences). For this reason this model is furthermore called 'the simplification learning model'. The training and the test datasets are the same as in the previous experiments, and also the same classifiers have been used. The results on the simplification learning model, as shown in Table 3, are lower, but nevertheless remarkable. This impact on the classifiers is expected, as this learning model uses only five features and, as can be seen from the previous analysis presented, there are other attributes which contribute in the original learning model.

The new learning model is able to classify the texts with accuracies between 87.64% to 88.89% on 10 fold cross-validation, and reaching up to 93.24% for the test dataset. These values may be considered an argument in favour of the simplification universal.

Table 2. Attributes Ranking Filters.

Chi-squared		**Information Gain**	
321.4558	InformationLoad	0.4367	InformationLoad
320.3795	Nouns	0.4207	Nouns
311.8009	Prepositions	0.4082	Prepositions
287.4316	LexicalRichness	0.3922	LexicalRichness
271.1993	CommonNouns	0.3391	GrammaticalWordsPerLexicalWords
258.2133	GrammaticalWordsPerLexicalWords	0.3387	CommonNouns
186.1927	VerbsPersThreeSingular	0.2319	VerbsPersThreeSingular
174.5388	Numerals	0.2304	Numerals
167.4469	GrammaticalWords	0.2081	GrammaticalWords
130.8715	SimpleSentences	0.17	SimpleSentences
59.031	VerbsIndicative	0.0743	Determiners
50.5388	Determiners	0.0738	VerbsIndicative
49.3664	Conjunctions	0.0639	Conjunctions
48.0164	Adverbs	0.0565	Adverbs
45.7126	ProperNouns	0.0537	ProperNouns
42.9458	VerbsParticiple	0.0507	VerbsParticiple
38.1205	VerbsGerund	0.0487	SentenceLenght
34.758	SentenceLenght	0.0441	VerbsGerund
29.3952	VerbsPersOnePlural	0.0327	VerbsPersOnePlural
28.0491	WordLength	0.0312	WordLength
24.395	Pronouns	0.0308	Pronouns
24.1608	Verbs	0.0294	Verbs
22.8642	VerbsPersTwoSingular	0.029	VerbsSubjonctive
21.9677	VerbsAux	0.0287	VerbsPersTwoSingular
18.515	VerbsSubjonctive	0.0249	VerbsAux
7.1135	AdjectivesSuperlative	0.0128	AdjectivesSuperlative
.....		.....	

Table 3. Classification Accuracies for the Simplification Learning Model

Classifier	*10-fold cross-validation*	*Test set*
Baseline	65.10%	66.89%
NaiveBayes	88.58%	85.81%
SVM	87.64%	87.84%
Jrip	88.42%	93.24%
J48	88.89%	96.62%

5 CONCLUSIONS AND FURTHER RESEARCH

This paper presents a new study on the investigation of universals of translations, for the Romanian language. A supervised learning approach is employed to identify the most informative features that characterise translations compared to non-translated texts. Additionally, an analysis of the impact on classification of the features previously proposed for the 'simplification universal' is conducted. The accuracies of the learning model in the categorisation task have outstanding values, and reach

up to 98.65% value on a randomly generated test dataset. All the classifiers register a decreased success rate when the simplification features are removed. However, the lowest result is still well above the chance level. The performance analysis on the classifiers' output reveals that the learning model relies highly on the following attributes: information load, lexical richness, proportion in texts of nouns, prepositions, grammatical words to lexical words, third person singular verbs, numerals and simple sentences. For future work, the inclusion of other features considered to stand for different translation universals in the learning model may bring different arguments towards their validity.

ACKNOWLEDGEMENTS

We would like to express our gratitude to members of staff at Research Group of Computational Linguistics from University of Wolverhampton, and in particular to Georgiana Puşcaşu, Dr. Constantin Orăsan, and Prof. Ruslan Mitkov for their invaluable suggestions, guidance and support. Also, we are thankful to Claudiu Mihăilă from University of Manchester for his prompt and constructive comments.

REFERENCES

1. Gellerstam, M.: Translationese in Swedish novels translated from English. Translation Studies in Scandinavia. Lund: CWK Gleerup (1986)
2. Baker, M.: Corpus linguistics and translation studies – implications and applications. In M. Baker, M.F..E.T.B., ed.: Text and Technology: In Honour of John Sinclair. Amsterdam & Philadelphia: John Benjamins (1993) 233–250
3. Baker, M.: Corpus-based translation studies: The challenges that lie ahead. In Somers, H., ed.: Terminology, LSP and Translation: Studies in Language Engineering in Honour of Juan C. Sager. Amsterdam & Philadelphia: John Benjamins (1996) 175–186
4. Borin, L., Prütz, K.: Thorough a dark glass: part of speech distribution in original and translated text. In Daelemans, W., Sima'an, K., Veenstra, J., Zavrel, J., eds.: Computational Linguistics in the Netherlands. Amsterdam: Rodopi (2001) 30–44
5. Hansen, S.: The Nature of Translated Text. Saarbrücken: Saarland University (2003)
6. Teich, E.: Cross-linguistic Variation in System and Text. Berlin: Mouton de Gruyter (2003)
7. Toury, G.: Descriptive Translation Studies and Beyond. Amsterdam: John Benjamins (1995)

8. Laviosa, S.: Corpus-based Translation Studies. Theory, Findings, Applications. Amsterdam & New York: Rodopi (2002)
9. Tymoczko, M.: Computerized corpora and the future of translation studies. Meta **43:4** (1998) 652–659
10. Bernardini, S., Zanettin, F.: When is a universal not a universal? In Mauranen, A., Kujamaki, P., eds.: Translation Universals. Do they exist? Amsterdam: Benjamins (2004) 51–62
11. Toury, G.: Probabilistic explanations in translation studies. welcome as they are, would they qualify as universals? In Mauranen, A., Kujamäki, P., eds.: Translation Universals: Do they exist? Amsterdam: John Benjamins (2004) 15–32
12. Chesterman, A.: A causal model for translation studies. In Olohan, M., ed.: Intercultural Faultlines. Research Models in Translation Studies I. Textual and Cognitive Aspects. St. Jerome (2000) 15–27
13. Resnik, P., Smith, N.: The web as a parallel corpus. Computational Linguistics **29(3)** (2003) 349–380
14. Goutte, C., Kurokawa, D., Isabelle, P.: Improving SMT by learning translation direction. In: Statistical Multilingual Analysis for Retrieval and Translation, Barcelona, Spain (2009)
15. Corpas, G.: Investigar con corpus en traducción: los retos de un nuevo paradigma. Frankfurt am Main, Berlin & New York: Peter Lang (2008)
16. Ilisei, I., Inkpen, D., Pastor, G., Mitkov, R.: Identification of translationese: A supervised learning approach. In Gelbukh, A., ed.: CICLing 2010, Lecture Notes in Computer Science 6008. Springer, Heidelberg (2010) 503–511
17. Baroni, M., Bernardini, S.: A new approach to the study of translationese: Machine-learning the difference between original and translated text. Literary and Linguistic Computing **21, 3** (2006) 259–274
18. Olohan, M., Baker, M.: Reported 'that' in translated English: Evidence for subconcious processes of explicitation? Across Languages and Culture **1:2** (2000) 141–158
19. Laviosa, S.: Core patterns of lexical use in a comparable corpus of English narrative prose. In Laviosa, S., ed.: The Corpus-Based Approach. Volume Special Issue of Meta. Montral: Les Presses de L'Universit de Montral (1998) 557–570
20. Corpas, G., Mitkov, R., Afzal, N., Pekar, V.: Translation universals: Do they exist? A corpus-based NLP study of convergence and simplification. In: Proceedings of the AMTA, Waikiki, Hawaii (2008)
21. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA data mining software: An update. SIGKDD Explorations **11(1)** (2009) 10–18
22. Witten, I.H., Frank, E.: Data Mining : Practical Machine Learning Tools and Techniques. Second edition edn. Morgan Kaufman (2005)
23. Olohan, M.: Introducing Corpora in Translation Studies. Routledge (2004)

24. Pym, A.: On toury's laws of how translators translate. In Pym, A., M.S., Simeoni, D., eds.: Beyond Descriptive Translation Studies. Benjamins (2008) 311–328
25. Tufiş, D., Ion, R., Ceauşu, A., Ştefănescu, D.: Racai's linguistic web services. In: Proceedings of the 6th Language Resources and Evaluation Conference - LREC 2008, Marrakech, Morocco, ELRA - European Language Ressources Association (2008)
26. Tufiş, D., Ştefănescu, D., Ion, R., Ceauşu, A.: RACAI's question answering system at QA@CLEF 2007. In Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D., eds.: Advances in Multilingual and Multimodal Information Retrieval (CLEF 2007), Lecture Notes in Computer Science. Volume 5152. Springer-Verlag (2008) 3284–3291
27. Ilisei, I., Inkpen, D., Corpas, G., Mitkov, R.: Towards simplification: A supervised learning approach. In: Proceedings of Machine Translation 25 Years On, London, United Kingdom. (2009)
28. Quinlan, J.R.: Induction of decision trees. Machine Learning **1** (1986) 81–106

IUSTINA ILISEI
RESEARCH INSTITUTE IN INFORMATION AND LANGUAGE PROCESSING,
UNIVERSITY OF WOLVERHAMPTON
WULFRUNA STREET, WOLVERHAMPTON WV1 1LY, UK
E-MAIL: <IUSTINA.ILISEI@GMAIL.COM>

DIANA INKPEN
SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING,
UNIVERSITY OF OTTAWA,
800, KING EDWARD STREET, OTTAWA, ON, K1N 6N5, CANADA
E-MAIL: <DIANA@SITE.UOTTAWA.CA>

Author Index

Reviewing Committee of the Volume

Sophia Ananiadou
Bogdan Babych
Sivaji Bandyopadhyay
Roberto Basili
Pushpak Bhattacharyya
Nicoletta Calzolari
Sandra Carberry
Kenneth Church
Dan Cristea
Silviu Cucerzan
Mona T. Diab
Alex Chengyu Fang
Anna Feldman
Daniel Flickinger
Alexander Gelbukh
Roxana Girju
Gregory Grefenstette
Iryna Gurevych
Nizar Habash
Sanda Harabagiu
Yasunari Harada
Ales Horak
Eduard Hovy
Nancy Ide
Diana Inkpen
Sylvain Kahane
Alma Kharrat
Adam Kilgarriff
Richard Kittredge
Sandra Kübler
Krister Lindén
Aurelio Lopez
Bernardo Magnini
Cerstin Mahlow
Christopher Manning
Yuji Matsumoto
Diana McCarthy
Rada Mihalcea
Ruslan Mitkov
Dunja Mladenic
Marie-Francine Moens
Dan Moldovan
Masaki Murata
Smaranda Muresan
Roberto Navigli
Kjetil Nørvåg
Kemal Oflazer
Constantin Orasan
Maria Teresa Pazienza
Ted Pedersen
Viktor Pekar
Anselmo Peñas
Irina Prodanof
James Pustejovsky
Fuji Ren
German Rigau
Fabio Rinaldi
Vasile Rus
Horacio Saggion
Franco Salvetti
Hassan Sawaf
Satoshi Sekine
Serge Sharoff
Grigori Sidorov
Thamar Solorio
Benno Stein
Vera Lúcia Strube De Lima
Jun Suzuki
Christoph Tillmann
George Tsatsaronis
Junichi Tsujii
Karin Verspoor

Manuel Vilares Ferro
Hai Feng Wang
Leo Wanner
Annie Zaenen

Additional Referees

Afzal, Naveed
Agerri, Rodrigo
Agustini, Alexandre
Al-Sabbagh, Rania
Alonso Alemany, Laura
Anderka, Maik
Annesi, Paolo
Aramaki, Eiji
Atserias, Jordi
Aziz, Wilker
Bach, Nguyen
Baisa, Vít
Bernstein, Jared
Bhaskar, Pinaki
Bisazza, Arianna
Blanco, Eduardo
Bohnet, Bernd
Bouayad-Agha, Nadjet
Cabrio, Elena
Carreras, Xavier
Chong, Miranda
Croce, Danilo
Das, Amitava
Das, Dipankar
Davis, Chris Irwin
De Belder, Jan
Decao, Diego
Dornescu, Iustin
Duh, Kevin
Frunza, Oana
Gasperin, Caroline
Giménez, Jesús
Gonzalez, Marco
Hagiwara, Masato
Hernández, Laritza
Higashinaka, Ryuichiro
Hoppe, Dennis
Iftene, Adrian
Ilisei, Iustina
Kimura, Yasutomo
King, Levi
Kolesnikova, Olga
Konstantinova, Natalia
Lavelli, Alberto
Ledeneva, Yulia
Li, Chen
Lipka, Nedim
Lopes, Lucelene
Maebo, Kanako
Maskey, Sameer
Mehdad, Yashar
Meyer, Christian M.
Mille, Simon
Mohler, Michael
Montes y Gomez, Manuel
Mulkar-Mehta, Rutu
Murakami, Koji
Naskar, Sudip Kumar
Nemcik, Vasek
Neumayer, Robert
Neverilova, Zuzana
Otoguro, Ryo
Pakray, Partha
Pal, Santanu
Piotrowski, Michael
Ponomareva, Natalia
Potthast, Martin
Ramampiaro, Heri

Razavi, Amir H.
Rello, Luz
Rigo, Stefan
Rocha-Junior, João B.
Rodrigo, Alvaro
Rudnick, Alex
Sarikaya, Ruhi
Schneider, Gerold
Seaward, Leanne
Sinha, Ravi
Smith, Amber
Sudoh, Katsuhito
Sun, Xiao
Temnikova, Irina
thornton, wren
Tokuhisa, Masato
Tonelli, Sara
Trandabat, Diana
Tratz, Stephen
Tsubota, Yasushi
Turmo, Jordi
Tymoshenko, Kateryna
Venkatapathy, Sriram
Vieira, Renata
Wacholder, Nina
Wonsever, Dina
Xavier, Clarissa
Xu, Jian-ming
Yuan, Caixia
Zhang, Rong
Zhao, Bing