Applying spelling error correction techniques for improving semantic role labelling

Erik Tjong Kim Sang
Informatics Institute
University of Amsterdam, Kruislaan 403
NL-1098 SJ Amsterdam, The Netherlands
erikt@science.uva.nl

Sander Canisius, Antal van den Bosch, Toine Bogers
ILK / Computational Linguistics and AI
Tilburg University, P.O. Box 90153,
NL-5000 LE Tilburg, The Netherlands
{S.V.M.Canisius,Antal.vdnBosch,
A.M.Bogers}@uvt.nl

1 Introduction

This paper describes our approach to the CoNLL-2005 shared task: semantic role labelling. We do many of the obvious things that can be found in the other submissions as well. We use syntactic trees for deriving instances, partly at the constituent level and partly at the word level. On both levels we edit the data down to only the predicted positive cases of verb-constituent or verb-word pairs exhibiting a verb-argument relation, and we train two next-level classifiers that assign the appropriate labels to the positively classified cases. Each classifier is trained on data in which the features have been selected to optimize generalization performance on the particular task. We apply different machine learning algorithms and combine their predictions.

As a novel addition, we designed an automatically trained post-processing module that attempts to correct some of the errors made by the base system. To this purpose we borrowed Levenshtein-distance-based correction, a method from spelling error correction to repair mistakes in sequences of labels. We adapted the method to our needs and applied it for improving semantic role labelling output. This paper presents the results of our approach.

2 Data and features

The CoNLL-2005 shared task data sets provide sentences in which predicate–argument relations have been annotated, as well as a number of extra annotations like named entities and full syntactic parses (Carreras and Màrquez, 2005). We have used the parses for generating machine learning instances for pairs of predicates and syntactic phrases. In principle each phrase can have a relation with each verb in the same sentence. However, in order to keep the number of instances at a reasonable number, we have only built instances for verb–phrase pairs when the phrase parent is an ancestor of the verb (400,128 training instances). A reasonable number of arguments are individual words; these do not match with phrase boundaries. In order to be able to label these, we have also generated instances for all pairs of verbs and individual words using the same constraint (another 542,217 instances). The parent node constraint makes certain that embedded arguments, which do not occur in these data sets, cannot be predicted by our approach.

Instances which are associated with verb–argument pairs receive the label of the argument as class while others in principle receive a NULL class. In an estimated 10% of the cases, the phrase boundaries assigned by the parser are different from those in the argument annotation. In case of a mismatch, we have always used the argument label of the first word of a phrase as the class of the corresponding instance. By doing this we attempt to keep the positional information of the lost argument in the training data. Both the parser phrase boundary errors as well as the parent node constraint restrict the number of phrases we can identify. The maximum recall score attainable with our phrases is 84.64% for the development data set.

We have experimentally evaluated 30 features based on the previous work in semantic role labelling (Gildea and Jurafsky, 2002; Pradhan et al., 2004; Xue and Palmer, 2004):

- **Lexical features** (5): predicate (verb), first phrase word, last phrase word and words immediately before and after the phrase.
- **Syntactic features** (14): part-of-speech tags (POS) of: first phrase word, last phrase word,

word immediately before phrase and word immediately after phrase; syntactic paths from word to verb: all paths, only paths for words before verb and only paths for words after verb; phrase label, label of phrase parent, subcategorisation of verb parent, predicate frame from PropBank, voice, head preposition for prepositional phrases and same parents flag.

- **Semantic features** (2): named entity tag for first phrase word and last phrase word.
- **Positional features** (3): position of the phrase with respect to the verb: left/right, distance in words and distance in parent nodes.
- **Combination features** (6): predicate + phrase label, predicate + first phrase word, predicate + last phrase word, predicate + first phrase POS, predicate + last phrase POS and voice + left/right.

The output of two parsers was available. We have briefly experimented with the Collins parses including the available punctuation corrections but found that our approach reached a better performance with the Charniak parses. We report only on the results obtained with the Charniak parses.

3 Approach

This section gives a brief overview of the three main components of our approach: machine learning, automatic feature selection and post-processing by a novel procedure designed to clean up the classifier output by correcting obvious misclassifications.

3.1 Machine learning

The core machine learning technique employed, is memory-based learning, a supervised inductive algorithm for learning classification tasks based on the k-nn algorithm. We use the TiMBL system (Daelemans et al., 2003), version 5.0.0, patch-2 with uniform feature weighting and random tiebreaking (options: -w 0 -R 911). We have also evaluated two alternative learning techniques. First, Maximum Entropy Models, for which we employed Zhang Le's Maximum Entropy Toolkit, version 20041229 with default parameters. Second, Support Vector Machines for which we used Taku Kudo's YamCha (Kudo and Matsumoto, 2003), with one-versus-all voting and option -V which enabled us to ignore predicted classes with negative distances.

3.2 Feature selection

In previous research, we have found that memory-based learning is rather sensitive to the chosen features. In particular, irrelevant or redundant features may lead to reduced performance. In order to minimise the effects of this sensitivity, we have employed bi-directional hill-climbing (Caruana and Freitag, 1994) for finding the features that were most suited for this task. This process starts with an empty feature set, examines the effect of adding or removing one feature and then starts a new iteration with the set associated with the best performance.

3.3 Automatic post-processing

Certain misclassifications by the semantic role-labelling system described so far lead to unlikely and impossible relation assignments, such as assigning two indirect objects to a verb where only one is possible. Our proposed classifier has no mechanism to detect these errors. One solution is to devise a post-processing step that transforms the resulting role assignments until they meet certain basic constraints, such as the rule that each verb may have only single instances of the different roles assigned in one sentence (Van den Bosch et al., 2004).

We propose an alternative automatically-trained post-processing method which corrects unlikely role assignments either by deleting them or by replacing them with a more likely one. We do not do this by knowledge-based constraint satisfaction, but rather by adopting a method for error correction based on Levenshtein distance (Levenshtein, 1965), or edit distance, as used commonly in spelling error correction. Levenshtein distance is a dynamically computed distance between two strings, accounting for the number of deletions, insertions, and substitutions needed to transform the one string into the other. Levenshtein-based error correction typically matches a new, possibly incorrect, string to a trusted lexicon of assumedly correct strings, finds the lexicon string with the smallest Levenshtein distance to the new string, and replaces the new string with the lexicon string as its likely correction. We implemented a roughly similar procedure. First, we generated a lexicon of semantic role labelling patterns of A0–A5 arguments of verbs on the basis of the entire training corpus and the PropBank verb frames. This lexicon contains entries such as *abandon* A0 V A1, and *categorize* A1 V A2 – a total of 43,033 variable-length role labelling patterns.

Next, given a new test sentence, we consider all of its verbs and their respective predicted role labellings, and compare each with the lexicon, searching the role labelling pattern with the same verb at the smallest Levenshtein distance (in case of an unknown verb we search in the entire lexicon). For example, in a test sentence the pattern *emphasize* A0 V A1 A0 is predicted. One closest lexicon item is found at Levenshtein distance 1, namely *emphasize* A0 V A1, representing a deletion of the final A0. We then use the nearest-neighbour pattern in the lexicon to correct the likely error, and apply all deletions and substitutions needed to correct the current pattern according to the nearest-neighbour pattern from the trusted lexicon. We do not apply insertions, since the post-processor module does not have the information to decide which constituent or word would receive the inserted label. In case of multiple possible deletions (e.g. in deleting one out of two A1s in *emphasize* A0 V A1 A1), we always delete the argument furthest from the verb.

4 Results

In order to perform the optimisation of the semantic role labelling process in a reasonable amount of time, we have divided it in four separate tasks: pruning the data for individual words and the data for phrases, and labelling of these two data sets. Pruning amounts to deciding which instances correspond with verb-argument pairs and which do not. This resulted in a considerable reduction of the two data sets: 47% for the phrase data and 80% for the word data. The remaining instances are assumed to define verb-argument pairs and the labelling tasks assign labels to them. We have performed a separate feature selection process in combination with the memory-based learner for each of the four tasks. First we selected the best feature set based on task accuracy. As soon as a working module for each of the tasks was available, we performed an extra feature selection process for each of the modules, optimising overall system $F_{\beta=1}$ while keeping the other three modules fixed.

	Words		Phrases	
Features	prune	label	prune	label
predicate	-0.04	**+0.05**	-0.25	-0.52
first word	**+0.38**	**+0.16**	-0.17	**+1.14**
last word	–	–	-0.01	**+1.12**
previous word	-0.06	**+0.02**	-0.05	**+0.74**
next word	-0.04	-0.08	**+0.44**	-0.16
part-of-speech first word	-0.01	-0.02	-0.07	-0.11
part-of-speech last word	–	–	-0.14	-0.45
previous part-of-speech	-0.12	-0.06	**+0.22**	-1.14
next part-of-speech	-0.08	-0.12	-0.01	-0.21
all paths	**+0.42**	**+0.10**	**+0.84**	**+0.75**
path before verb	+0.00	-0.02	+0.00	**+0.27**
path after verb	-0.01	-0.01	-0.01	-0.06
phrase label	-0.01	-0.02	**+0.13**	-0.02
parent label	**+0.03**	-0.02	-0.03	+0.00
voice	+0.02	-0.04	-0.04	**+1.85**
subcategorisation	-0.01	+0.00	-0.02	+0.03
PropBank frame	-0.12	-0.03	-0.16	**+1.04**
PP head	+0.00	+0.00	-0.06	**+0.08**
same parents	-0.02	-0.01	**+0.03**	-0.05
named entity first word	+0.00	+0.00	+0.05	-0.11
named entity last word	–	–	-0.04	-0.12
absolute position	**+0.00**	+0.00	+0.00	-0.02
distance in words	**+0.34**	**+0.04**	**+0.16**	-0.96
distance in parents	-0.02	-0.02	**+0.06**	-0.04
predicate + label	-0.05	-0.07	-0.22	-0.47
predicate + first word	-0.05	+0.00	**+0.13**	**+0.97**
predicate + last word	–	–	-0.03	**+0.08**
predicate + first POS	-0.05	-0.06	-0.20	-0.50
predicate + last POS	–	–	-0.13	-0.40
voice + position	+0.02	-0.04	-0.05	-0.04

Table 1: Effect of adding a feature to the best feature sets when memory-based learning is applied to the development set (overall $F_{\beta=1}$). The process consisted of four tasks: pruning data sets for individual words and phrases, and labelling these two data sets. Selected features are shown in **bold**. Unfortunately, we have not been able to use all promising features.

The effect of the features on the overall performance can be found in Table 1. One feature (syntactic path) was selected in all four tasks but in general different features were required for optimal performance in the four tasks. Changing the feature set had the largest effect when labelling the phrase data. We have applied the two other learners, Maximum Entropy Models and Support Vector Machines to the two labelling tasks, while using the same features as the memory-based learner. The performance of the three systems on the development data can be found in Table 3. Since the systems performed differently we have also evaluated the performance of a combined system which always chose the majority class assigned to an instance and the class of the strongest system (SVM) in case of a three-way tie. The combined system performed slightly better than the best

	Precision	Recall	$F_{\beta=1}$
Development	76.79%	70.01%	73.24
Test WSJ	79.03%	72.03%	75.37
Test Brown	70.45%	60.13%	64.88
Test WSJ+Brown	77.94%	70.44%	74.00

Test WSJ	Precision	Recall	$F_{\beta=1}$
Overall	79.03%	72.03%	75.37
A0	85.65%	81.73%	83.64
A1	76.97%	71.89%	74.34
A2	71.07%	58.20%	63.99
A3	69.29%	50.87%	58.67
A4	75.56%	66.67%	70.83
A5	100.00%	40.00%	57.14
AM-ADV	64.36%	51.38%	57.14
AM-CAU	75.56%	46.58%	57.63
AM-DIR	48.98%	28.24%	35.82
AM-DIS	81.88%	79.06%	80.45
AM-EXT	87.50%	43.75%	58.33
AM-LOC	62.50%	50.96%	56.15
AM-MNR	64.52%	52.33%	57.78
AM-MOD	96.76%	97.64%	97.20
AM-NEG	97.38%	96.96%	97.17
AM-PNC	45.98%	34.78%	39.60
AM-PRD	50.00%	20.00%	28.57
AM-REC	0.00%	0.00%	0.00
AM-TMP	80.52%	70.75%	75.32
R-A0	81.47%	84.38%	82.89
R-A1	74.00%	71.15%	72.55
R-A2	60.00%	37.50%	46.15
R-A3	0.00%	0.00%	0.00
R-A4	0.00%	0.00%	0.00
R-AM-ADV	0.00%	0.00%	0.00
R-AM-CAU	100.00%	25.00%	40.00
R-AM-EXT	100.00%	100.00%	100.00
R-AM-LOC	86.67%	61.90%	72.22
R-AM-MNR	33.33%	33.33%	33.33
R-AM-TMP	64.41%	73.08%	68.47
V	97.36%	97.36%	97.36

Table 2: Overall results (top) and detailed results on the WSJ test (bottom).

individual system.

5 Conclusion

We have presented a machine learning approach to semantic role labelling based on full parses. We have split the process in four separate tasks: pruning the data bases of word-based and phrase-based examples down to only the positive verb-argument cases, and labelling the two positively classified data sets. A novel automatic post-processing procedure based on spelling correction, comparing to a trusted lexicon of verb-argument patterns from the training material, was able to achieve a performance increase by correcting unlikely role assignments.

Learning algorithm	Precision	Recall	$F_{\beta=1}$
without post-processing:			
Maximum Entropy Models	70.78%	70.03%	70.40
Memory-Based Learning	70.70%	69.85%	70.27
Support Vector Machines	75.07%	69.15%	71.98
including post-processing:			
Maximum Entropy Models	74.06%	69.84%	71.89
Memory-Based Learning	73.84%	69.88%	71.80
Support Vector Machines	77.75%	69.11%	73.17
Combination	76.79%	70.01%	73.24

Table 3: Effect of the choice of machine learning algorithm, the application of Levenshtein-distance-based post-processing and the use of system combination on the performance obtained for the development data set.

Acknowledgements

This research was funded by NWO, the Netherlands Organisation for Scientific Research, and by SenterNovem IOP-MMI.

References

X. Carreras and L. Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of CoNLL-2005*. Ann Arbor, MI, USA.

R. Caruana and D. Freitag. 1994. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36, New Brunswick, NJ, USA. Morgan Kaufman.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2003. TiMBL: Tilburg memory based learner, version 5.0, reference guide. ILK Technical Report 03-10, Tilburg University.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

T. Kudo and Y. Matsumoto. 2003. Fast methods for kernel-based text analysis. In *Proceedings of ACL-2003*. Sapporo, Japan.

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

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the HLT/NAACL 2004*. Boston, MA.

A. van den Bosch, S. Canisius, W. Daelemans, I Hendrickx, and E. Tjong Kim Sang. 2004. Memory-based semantic role labeling: Optimizing features, algorithm, and output. In *Proceedings of the CoNLL-2004*, Boston, MA, USA.

N. Xue and M. Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of EMNLP-2004*. Barcelona, Spain.