Semantic Role Labeling using Dependency Syntax

Anders Björkelund
Department of Computer Science
Lund University, Sweden
dc@m0m0.org

Love Hafdell
Department of Computer Science
Lund University, Sweden
love_hafdell@hotmail.com

Abstract

This document gives a brief introduction to the topic of Semantic Role Labeling using Dependency Syntax. We also describe a system that has been developed and tested on a corpus from the CoNLL-2008[1] shared task. We evaluate the system and give a short discussion on further improvements. Our results are reasonably good compared to those reached during the shared task.

1 Introduction

Semantic role labeling (SRL) involves identifying events, their participants and how they relate to each other, i.e. *who* did *what* to *whom*. Additionally we are also interested in answering questions such as *how*, *when* and *where*. The event is described by a *predicate*, a verb or a noun. Participants and adjuncts describing the event are referred to as *arguments*.

Finding a generic way to describe all kinds of events is far from trivial and there is a number of semantic lexicons such as FrameNet, VerbNet and PropBank. The lexicons define one or more senses for a certain predicate (Figure 1). Most predicates only have one sense, but in rare cases it can range up to about a dozen. Furthermore, each lexicon assigns every sense a number of possible *semantic roles*, a pre-defined set of relations that might hold between the predicate and potential arguments.

We have developed a system to perform automatic SRL using machine-learning techniques. The training and testing data, as well as an evaluation script, derives from the CoNLL-2008 shared task. This allowed us to compare our results with the results of the participants in the shared task. The vastness of the shared task combined with the short time span of our project, however, prevented us from reaching scores at the level of those that were reached during the shared task.

In the next section we mention some previous work and describe the CoNLL-2008 shared task. We go on and mention

John keeps quiet

John keeps a diary

Figure 1: Keep appears in two different senses

2 Previous Work

Labeling of semantic roles is an important step towards solving major Natural Language Processing (NLP) problems such as template filling and question answering. It has also been shown to improve document categorization (Persson, 2008).

Using statistical methods to solve NLP problems requires hand-annotated corpora of significant size. This was introduced with resources such as FrameNet and PropBank. The famous work by Gildea and Jurafsky (2002) has largely influenced the field of automatic SRL in terms of feature selection and the use of statistical methods (Moschitti et al., 2008). In recent years SRL has received much focus during conferences such as CoNLL-2005,

[1] http://www.cnts.ua.ac.be/conll2008/

SemEval-2007 and CoNLL-2008.
The consensus seems to be that SRL should be made on top of a syntactic representation. This representation has usually been constituent based, as opposed to the the dependency based used in the CoNLL-2008. The choice of syntactic representation can be argued, though recently it has been shown that the two perform more or less equally well when applied to SRL (Johansson, 2008).

2.1 CoNLL-2008 shared task

The CoNLL-2008 shared task involved not only SRL, but also syntactic parsing. The corpus used derives from the Penn Treebank that had been converted from a constituent to a dependency based syntactic representation. The first part of the task consisted of parsing the syntactic dependencies, and the second, which is what we have done during this project, was the SRL step. In addition to ordinary training and testing data, supplementary files with syntactic dependencies generated by a state-of-the-art dependency parser were provided. The lexicons used were PropBank[2] for verbs and NomBank[3] for nouns. Two test sets were provided, the Wall Street Journal (WSJ) test set, which belongs to the same domain as the training set, and the Brown test set, which did not (Surdeanu et al., 2008).

3 Labeling of Semantic Roles

An example of a semantic tree describing a sentence is given in Figure 2. The predicate of this sentence is *come* and it's arguments are *test*, *may* and *today*. The sense of the predicate is denoted by the .01 suffix, which is defined in the PropBank lexicon. The PropBank frame describing come.01 refers to A1 as the "entity in motion/comer". The other arguments are not core arguments of the predicate but can be considered adjuncts in a more general sense. The AM-TMP label denotes the temporal information describing the event and the AM-MOD label in which manner the event occurs. In total there are 53 such labels in the corpus we have been using.

[2]http://verbs.colorado.edu/ mpalmer/projects/ace.html
[3]http://nlp.cs.nyu.edu/meyers/NomBank.html

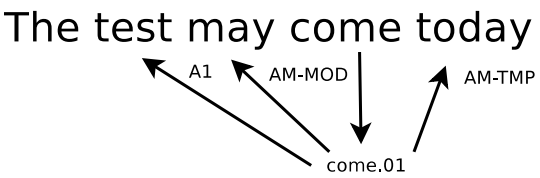

Figure 2: Semantic tree of a sentence

3.1 Verbs and nouns

Aside from verb predicates, such as in Figure 2, there are also noun predicates. For instance in the sentence "John is the king of Sweden", king is a predicate, and it's arguments are John and Sweden. As we've mentioned before, the lexicon used in CoNLL-2008 for noun predicates is called NomBank. It was developed to be consistent with PropBank in terms of argument labels. Unfortunately it's not always intuitive what labels to use, and seeing as we have solved the problem by training classifiers on a pre-annotated training set, the actual labels is more a matter of convention than intuition.

3.2 The pipeline approach

Labeling semantic roles, ie finding a tree such as in Figure 2, is normally broken down into four steps:
Predicate Identification, identifying the predicate come;
Predicate Disambiguation, resolving the .01 suffix of the come predicate;
Argument Identification, noting that test, may and today are arguments;
Argument Classification, assigning the labels to the arcs in the tree.

4 Constructed System

We have implemented a system consisting of four modules, one for each step of the pipeline. The system was implemented in Java and used the LibLinear package (Fan et al., 2008) to create linear L2-regularized logistic classifiers. In all steps but the predicate disambiguation step, two classifiers were trained, one for verbs and one for nouns. Disambiguation required one classifier for each lemma, as there is no consistency over predicates when it comes to sense labels.

4.1 Technical difficulties

The training set consists of almost one million lines (one word per line), making up about 40,000 sentences. This requires plenty of memory both in terms of processing the data within our program, as well as training the classifiers. The choice of programming language and data structures may not have been optimal in this sense, although we tried a couple of different data structures and ideas to model the data. Eventually we settled with using the standard Java API and the primitive data types of Java. We were also given access to a fast computer with 32GB of internal memory. This turned out to be crucial to the development as it speeded up execution times significantly and we did not have to worry about any memory limitations. In this setting training of classifiers took roughly 3 hours, and classification of the larger training set roughly 10 minutes.

4.2 Classifiers

A benefit of using LibLinear as opposed to popular Support Vector Machine classifiers is that it allows significantly larger featuresets. A downside of linear classifiers is that they can not detect xor relations, which in turn caused us to make joint features out of existing ones by taking the cartesian product between two features. The features used total to 22 but varies from step to step and also between noun and verb classifiers, roughly 10 are used for each classifier. Yet the vectors representing the data have a dimensions up to about 2.5 millions. Most of the features used were inspired by the work previously done at the institution (Johansson and Nugues, 2007).

4.3 Dependency syntax

The "state-of-the-art" dependency parser that were used to provide the dependencies with the CoNLL-2008 had a Labeled Attachment Score (LAS) of 85% and 77% on the WSJ and Brown test sets, respectively. This was outperformed by most of the participants in the shared task. This implies that we had a worse conditions to start the SRL step. As there is a gold standard test set for comparison, we could allow ourselves to start with correct dependencies (100% LAS) since the main objective of this project was SRL and not syntactic parsing. We present the results of both cases in the next section.

4.4 Results

Evaluating results is far from trivial. Thankfully we had the CoNLL-2008 evaluation script which not only computes a number of measures, but also gives detailed information on what's right and what's wrong. There is a number of measures, the LAS for dependencies mentioned in the previous section being one. For semantics the two primary measures are the labeled and unlabeled F1 measures. Where the latter one refers to getting the arcs in the semantic tree (cf. Figure 1), and the labeled score takes the actual labels into consideration too.

Using state-of-the art dependencies

	Labeled F1	Unlabeled F1
WSJ	74.35%	83.46%
Brown	62.12%	78.94%

Using gold standard dependencies

	Labeled F1	Unlabeled F1
WSJ	78.25%	87.64%
Brown	67.65%	84.70%

5 Conclusion and Discussion

We see the expected drop in scores between the two tables. We must admit we were pleasantly surprised comparing these with those of the CoNLL-2008 shared task. There were 25 participating submissions (some of which were submitted post-deadline) in the original task, and the median labeled F1 are about 70% and 60% for the WSJ and the Brown test sets, respectively. Although our results were slightly better than that, the top scoring system was significantly better, which had labeled F1 scores of 83.05% and 69.85%, respectively. This is even better than what we acheived by using the gold standard dependencies.

We see that reaching better scores is not primarily a matter of what dependencies we use, but the method. The pipeline method we have used performs each step independently without taking the joint structure of the sentence into consideration. Using a joint model has been shown to improve results (Toutanova et al., 2008) and was also used by participants, including the top system, in the shared task (Johansson and Nugues, 2008).

The features used could be investigated more thoroughly. We did not investigate the impact of individual features, and additional features than those that we used have been suggested (Johansson and Nugues, 2007). Furthermore we suspect that some features we used were probably not implemented optimally. For instance, one feature used is the voice of verbs (ie. active or passive) in the case of verb predicates. To determine whether a verb is in active or passive voice is not trivial and we might have missed a few cases of passive.

All in all there's more work to be done to reach state-of-the-art results, and a long way to go to achieve even better. The NLP community is continuously working on the topic of SRL. This will also be the topic of the next CoNLL shared task in 2009.

References

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. *LIBLINEAR: A Library for Large Linear Classification*. Journal of Machine Learning Research 9(2008)

Daniel Gildea, Daniel Jurafsky. 2002. *Automatic Labeling of Semantic Roles*. Computational Linguistics, Volume 28, Number 3

Richard Johansson, Pierre Nugues. 2007. *Syntactic Representations Considered for Frame-Semantic Analysis*. Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories.

Richard Johansson, Pierre Nugues. 2008. *The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies*. Proceedings of the 12th Conference on Computational Natural Language Learning.

Richard Johansson. 2008. *Dependency-based Semantic Analysis of Natural-language Text*. Doctoral dissertation, Department of Computer Science, Lund University.

Alessandro Moschitti, Daniele Pighin, Roberto Basili 2008. *Tree Kernels for Semantic Role Labeling*. Computational Linguistics, Volume 34, Number 2

Jacob Persson. 2008. *Textkategorisering med predikat-argumentstrukturer*. Masters thesis, Lund University

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, Joakim Nivre. 2008. *The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies*. CoNLL 2008: Proceedings of the 12th Conference on Computational Natural Language Learning.

Kristina Toutanova, Aria Haghighi, Christopher D. Manning. 2008. *A Global Joint Model for Semantic Role Labeling*. Computational Linguistics, Volume 34, Number 2