A High-Performance Syntactic and Semantic Dependency Parser

Anders Björkelund† **Bernd Bohnet**‡ **Love Hafdell**† **Pierre Nugues**†
†Department of Computer science
Lund University
anders.bjorkelund@cs.lth.se
love.hafdell@cs.lth.se
pierre.nugues@cs.lth.se
‡Institute for Natural Language Processing
University of Stuttgart
bohnet@ims.uni-stuttgart.de

Abstract

This demonstration presents a high-performance syntactic and semantic dependency parser. The system consists of a pipeline of modules that carry out the tokenization, lemmatization, part-of-speech tagging, dependency parsing, and semantic role labeling of a sentence. The system's two main components draw on improved versions of a state-of-the-art dependency parser (Bohnet, 2009) and semantic role labeler (Björkelund et al., 2009) developed independently by the authors.

The system takes a sentence as input and produces a syntactic and semantic annotation using the CoNLL 2009 format. The processing time needed for a sentence typically ranges from 10 to 1000 milliseconds. The predicate–argument structures in the final output are visualized in the form of segments, which are more intuitive for a user.

1 Motivation and Overview

Semantic analyzers consist of processing pipelines to tokenize, lemmatize, tag, and parse sentences, where all the steps are crucial to their overall performance. In practice, however, while code of dependency parsers and semantic role labelers is available, few systems can be run as standalone applications and even fewer with a processing time per sentence that would allow a user interaction, i.e. a system response ranging from 100 to 1000 milliseconds.

This demonstration is a practical semantic parser that takes an English sentence as input and produces syntactic and semantic dependency graphs using the CoNLL 2009 format. It builds on lemmatization and POS tagging preprocessing steps, as well as on two systems, one dealing with syntax and the other with semantic dependencies that reported respectively state-of-the-art results in the CoNLL 2009 shared task (Bohnet, 2009; Björkelund et al., 2009). The complete system architecture is shown in Fig. 1.

The dependency parser is based on Carreras's algorithm (Carreras, 2007) and second order spanning trees. The parser is trained with the margin infused relaxed algorithm (MIRA) (McDonald et al., 2005) and combined with a hash kernel (Shi et al., 2009). In combination with the system's lemmatizer and POS tagger, this parser achieves an average labeled attachment score (LAS) of 89.88 when trained and tested on the English corpus of the CoNLL 2009 shared task (Surdeanu et al., 2008).

The semantic role labeler (SRL) consists of a pipeline of independent, local classifiers that identify the predicates, their senses, the arguments of the predicates, and the argument labels. The SRL module achieves an average labeled semantic F1 of 80.90 when trained and tested on the English corpus of CoNLL 2009 and combined with the system's preprocessing steps and parser.

2 The Demonstration

The demonstration runs as a web application and is available from a server located at http://

[*] Authors are listed in alphabetical order.

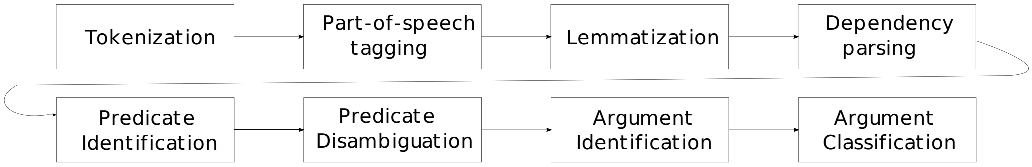
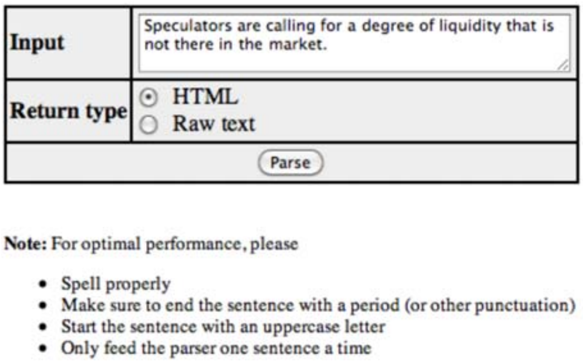


Figure 1: The overall system architecture.



Speculators are calling for a degree of liquidity that is not there in the market.

● HTML

Note: For optimal performance, please

- Spell properly
- Make sure to end the sentence with a period (or other punctuation)
- Start the sentence with an uppercase letter
- Only feed the parser one sentence a time

Figure 2: The input window, where the user entered the sentence *Speculators are calling for a degree of liquidity that is not there in the market.* Clicking on the **Parse** button starts the parser.

`barbar.cs.lth.se:8081/`. Figure 2 shows the input window, where the user can write or paste a sentence, here *Speculators are calling for a degree of liquidity that is not there in the market.*

Figure 3 shows the system output. It visualizes the end results as a list of predicates and their respective arguments in the form of colored segments. It also details the analysis as tabulated data using the CoNLL 2009 format (Surdeanu et al., 2008; Hajič et al., 2009), where the columns contain for each word, its form, lemma, POS tag, syntactic head, grammatical function, whether it is a predicate, and, if yes, the predicate sense. Then, columns are appended vertically to the table to identify the arguments of each predicate (one column per predicate). Figure 3 shows that the sentence contains two predicates, *call.03* and *degree.01* and the two last columns of the table show their respective arguments. Clicking on a predicate in the first column shows the description of its arguments in the PropBank or NomBank dictionaries. For *call.03*, this will open a new window that will show that Arg0 is the *demander*, Arg1, the *thing being demanded*, and Arg2, the *demandee*.

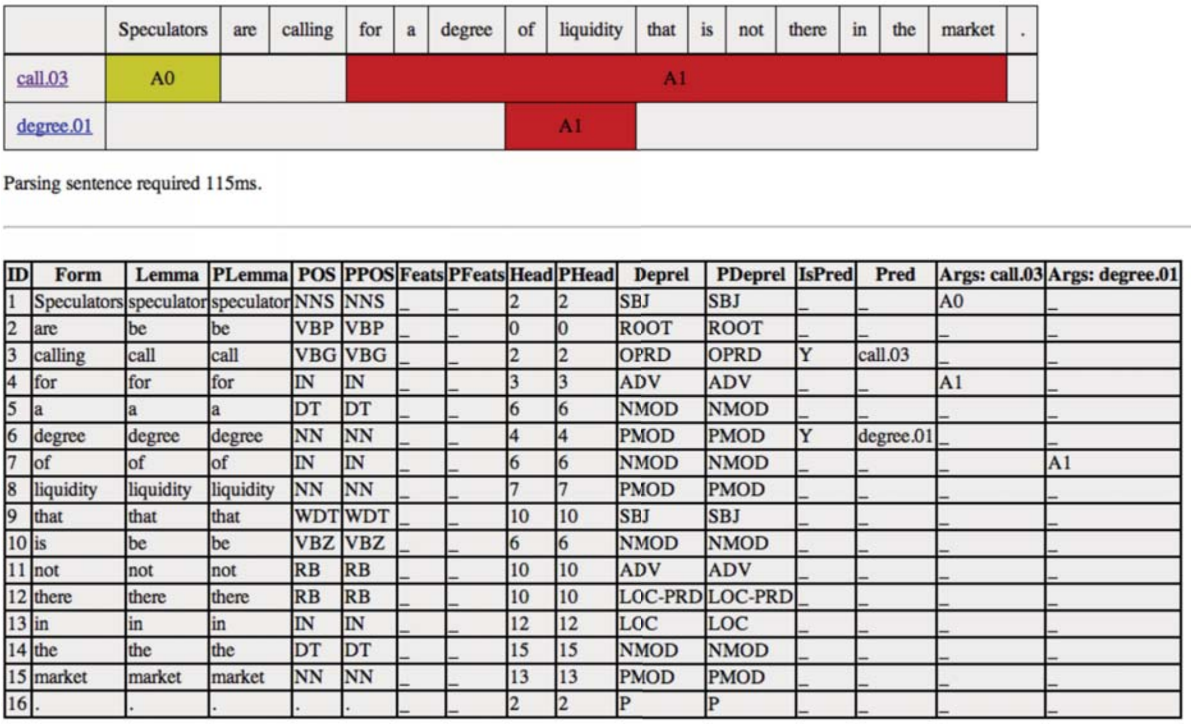
3 Preprocessing Steps

The preprocessing steps consist of the tokenization, lemmatization, and part-of-speech tagging of the input sentence. We use first OpenNLP[1] to tokenize the sentence. Then, the lemmatizer identifies the lemmas for each token and the tagger assigns the part-of-speech tags. The lemmatizer and the tagger use a rich feature set that was optimized for all languages of the CoNLL 2009 shared task (Hajič et al., 2009). Our lemmatizer uses the shortest edit script (SES) between the lemmas and the forms and we select a script within an SES list using a MIRA classifier (Chrupala, 2006). The English lemmatizer has an accuracy of 99.46. This is 0.27 percentage point lower than the predicted lemmas of the English corpus in CoNLL 2009, which had an accuracy of 99.73. The German lemmatizer has an accuracy of 98.28. The accuracy of the predicted lemmas in the German corpus was 68.48. The value is different because some closed-class words are annotated differently (Burchardt et al., 2006). We also employed MIRA to train the POS classifiers. Compared to the predicted POS tags in the shared task, we could increase the accuracy by 0.15 from 97.48 to 97.63 for English and by 1.55 from 95.68 to 97.23 for German.

4 Dependency Parsing

The dependency parser of this demonstration is a further development of Carreras (2007) and Johansson and Nugues (2008). We adapted it to account for the multilingual corpus of the CoNLL 2009 shared task – seven languages – and to improve the speed of the computationally expensive higher order decoder (Bohnet, 2009). The parser

[1] `http://opennlp.sourceforge.net/`



	Speculators	are	calling	for	a	degree	of	liquidity	that	is	not	there	in	the	market	.
call.03	A0			A1												
degree.01							A1									

Parsing sentence required 115ms.

ID	Form	Lemma	PLemma	POS	PPOS	Feats	PFeats	Head	PHead	Deprel	PDeprel	IsPred	Pred	Args: call.03	Args: degree.01
1	Speculators	speculator	speculator	NNS	NNS	_	_	2	2	SBJ	SBJ	_	_	A0	_
2	are	be	be	VBP	VBP	_	_	0	0	ROOT	ROOT	_	_	_	_
3	calling	call	call	VBG	VBG	_	_	2	2	OPRD	OPRD	Y	call.03	_	_
4	for	for	for	IN	IN	_	_	3	3	ADV	ADV	_	_	A1	_
5	a	a	a	DT	DT	_	_	6	6	NMOD	NMOD	_	_	_	_
6	degree	degree	degree	NN	NN	_	_	4	4	PMOD	PMOD	Y	degree.01	_	_
7	of	of	of	IN	IN	_	_	6	6	NMOD	NMOD	_	_	_	A1
8	liquidity	liquidity	liquidity	NN	NN	_	_	7	7	PMOD	PMOD	_	_	_	_
9	that	that	that	WDT	WDT	_	_	10	10	SBJ	SBJ	_	_	_	_
10	is	be	be	VBZ	VBZ	_	_	6	6	NMOD	NMOD	_	_	_	_
11	not	not	not	RB	RB	_	_	10	10	ADV	ADV	_	_	_	_
12	there	there	there	RB	RB	_	_	10	10	LOC-PRD	LOC-PRD	_	_	_	_
13	in	in	in	IN	IN	_	_	12	12	LOC	LOC	_	_	_	_
14	the	the	the	DT	DT	_	_	15	15	NMOD	NMOD	_	_	_	_
15	market	market	market	NN	NN	_	_	13	13	PMOD	PMOD	_	_	_	_
16	.	.	.	.	.	_	_	2	2	P	P	_	_	_	_

Figure 3: The output window. The predicates and their arguments are shown in the upper part of the figure, respectively *call.03* with *A0* and *A1* and *degree.01* with *A1*, while the results in the CoNLL 2008 format are shown in the lower part.

reached the best accuracies in CoNLL 2009 for English and German, and was ranked second in average over all the languages in the task.

The parser in this demonstration is an enhancement of the CoNLL 2009 version with a *hash kernel*, a parallel parsing algorithm, and a parallel feature extraction to improve the accuracy and parsing speed. The hash kernel enables the parser to reach a higher accuracy. The introduction of this kernel entails a modification of MIRA, which is simple to carry out: We replaced the feature-index mapping that mapped the features to indices of the weight vector by a random function. Usually, the feature-index mapping in a support vector machine has two tasks: It maps the features to an index in the weight vector and filters out the features not collected in the first step. The parser is about 12 times faster than a baseline parser without hash kernel and without parallel algorithms. The parsing time is about 0.077 seconds per sentence in average for the English test set.

5 Semantic Role Labeling Pipeline

The pipeline of classifiers used in the semantic role labeling consists of four steps: predicate identification, predicate disambiguation, argument identification, and argument classification, see Fig. 1. In each step, we used different classifiers for the nouns and the verbs. We build all the classifiers using the L2-regularized linear logistic regression from the LIBLINEAR package (Fan et al., 2008). To speed up processing, we disabled the reranker used in the CoNLL 2009 system (Björkelund et al., 2009).

Predicate Identification is carried out using a binary classifier that determines whether a noun or verb is a predicate or not.

Predicate Disambiguation is carried out for all the predicates that had multiple senses in the training corpus. We trained one classifier per lemma. For lemmas that could be both a verb or a noun (e.g. *plan*), we trained one classifier per part of speech. We considered lemmas with a unique observed sense as unambiguous.

Argument Identification and Classification. Similarly to the two previous steps, a binary classifier first identifies the arguments and then a multiclass classifier assigns them a label. In both steps, we used separate models for the nouns and the verbs.

Features. For the predicate identification, we used the features suggested by Johansson and Nugues (2008). For the other modules of the pipeline, we used the features outlined in Björkelund et al. (2009). The feature sets were originally selected using a greedy forward procedure. We first built a set of single features and, to improve the separability of our linear classifiers, we paired features to build bigrams.

6 Results and Discussion

The demonstration system implements a complete semantic analysis pipeline for English, where we combined two top-ranked systems for syntactic and semantic dependency parsing of the CoNLL 2009 shared task. We trained the classifiers on the same data sets and we obtained a final semantic F1 score of 80.90 for the full system. This score is lower than the best scores reported in CoNLL 2009. It is not comparable, however, as the predicates had then been manually marked up. Our system includes a predicate identification stage to carry out a fully automatic analysis. This explains a part of the performance drop. To provide comparable figures, we replaced the predicate identification classifier with an oracle reading the gold standard. We reached then a score of 85.58. To reach a higher speed and provide an instantaneous response to the user (less than 1 sec.), we also removed the global reranker from the pipeline which accounts for an additional loss of about 1.2 percentage point. This would put the upper-bound semantic F1 value to about 86.80, which would match the CoNLL 2009 top figures.

Acknowledgments. The research leading to these results has received funding from the European community's seventh framework program FP7/2007-2013, challenge 2, cognitive systems, interaction, robotics, under grant agreement No 230902—ROSETTA.

References

Björkelund, Anders, Love Hafdell, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of CoNLL-2009*.

Bohnet, Bernd. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of CoNLL-09*.

Burchardt, Aljoscha, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. 2006. The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of the 5th LREC-2006*.

Carreras, Xavier. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of CoNLL-2007*.

Chrupala, Grzegorz. 2006. Simple data-driven context-sensitive lemmatization. In *Proceedings of SEPLN*.

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Hajič, Jan, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL-2009*.

Johansson, Richard and Pierre Nugues. 2008. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings CoNLL-2008*.

McDonald, Ryan, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL-2005*.

Shi, Qinfeng, JamesPetterson, Gideon Dror, John Langford, Alex Smola, and S.V.N. Vishwanathan. 2009. Hash kernels for structured data. *Journal of Machine Learning*, 15(1):143–172.

Surdeanu, Mihai, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL–2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL–2008*.