A Simple Generative Pipeline Approach to Dependency Parsing and Semantic Role Labeling

Daniel Zeman
Ústav formální a aplikované lingvistiky
Univerzita Karlova v Praze
Malostranské náměstí 25, Praha, CZ-11800, Czechia
zeman@ufal.mff.cuni.cz

Abstract

We describe our CoNLL 2009 Shared Task system in the present paper. The system includes three cascaded components: a generative dependency parser, a classifier for syntactic dependency labels and a semantic classifier. The experimental results show that the labeled macro F1 scores of our system on the joint task range from 43.50% (Chinese) to 57.95% (Czech), with an average of 51.07%.

1 Introduction

The CoNLL 2009 shared task is an extension of the tasks addressed in previous years: unlike the English-only 2008 task, the present year deals with seven languages; and unlike 2006 and 2007, semantic role labeling is performed atop the surface dependency parsing.

We took part in the closed challenge of the joint task.[1] The input of our system contained gold standard lemma, part of speech and morphological features for each token. Tokens which were considered *predicates* were marked in the input data. The system was required to find the following information:

- parent (syntactic dependency) for each token
- label for each syntactic dependency (token)
- label for every predicate
- for every token (predicate or non-predicate) *A* and every predicate *P* in the sentence, say whether there is a semantic relation between *P* and *A* (*A* is an argument of *P*) and if so, provide a label for the relation (role of the argument)

The organizers of the shared task provided training and evaluation data (Hajič et al., 2006; Surdeanu et al., 2008; Burchardt et al., 2006; Taulé et al., 2008; Kawahara et al., 2002; Xue and Palmer, 2009) converted to a uniform CoNLL Shared Task format.

2 System Description

The system is a sequence of three components: a surface syntactic parser, a syntactic tagger that assigns labels to the syntactic dependencies and a semantic classifier (labels both the predicates and the roles of their arguments). We did not attempt to gain advantage from training a joint classifier for all the subtasks. We did not have time to do much beyond putting together the basic infrastructure. The components 2 and 3 are thus fairly primitive.

2.1 Surface Dependency Parser

We use the parser described by Zeman (2004). The parser takes a generative approach. It has a model of dependency statistics in which a dependency is

[1] For more details on the two tasks and challenges, see Hajič et al. (2009).

Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task, pages 120–125,
Boulder, Colorado, June 2009.

specified by the lemma and tag of the parent and the child nodes, by direction (left or right) and adjacency. The core of the algorithm can be described as repeated greedy selecting of best-weighted *allowed* dependencies and adding them to the dependency tree.

There are other components which affect the dependency selection, too. They range from supporting statistical models to a few hard-coded rules. However, some features of the parser are designed to work with Czech, or even with the Prague Dependency Treebank. For instance, there is a specialized model for coordinative constructions. The model itself is statistical but it depends on the PDT annotation guidelines in various ways. Most notably, the training component recognizes coordination by the `Coord` dependency label, which is not present in other treebanks. Other rules (e.g. the constraints on the set of allowed dependencies) rely on correct interpretation of the part-of-speech tags.

In order to make the parser less language-dependent in the multilingual environment of the shared task, we disabled most of the abovementioned treebank-bound features. Of course, it led to decreased performance on the Czech data.[2]

2.2 Assignment of Dependency Labels

The system learns surface dependency labels as a function of the part-of-speech tags and features of the parent and the child node. Almost no back-off is applied. The most frequent label for the given pair of tags (and feature structures) is always selected. If the pair of tags is unknown, the label is based on the features of the child node, and if it is unknown, too, the most frequent label of the training data is selected.

Obviously, both the training and the labeling procedures have to know the dependencies. Gold standard dependencies are examined during training while parser-generated dependencies are used for real labeling.

2.3 Semantic Classifier

The semantic component solves several tasks. First, all predicates have to be labeled. Tokens that are considered predicates in the particular treebank are marked on input, so this is a simple classification problem. Again, we took the path of least resistance and trained the PRED labels as a function of gold-standard lemmas.

Second, we have to find semantic dependencies. Any token (predicate or not) can be the argument of one or more predicates. These relations may or may not be parallel to a syntactic dependency. For each token, we need to find out 1. which predicates it depends on, and 2. what is the label of its semantic role in this relation?

The task is complex and there are apparently no simple solutions to it. We learn the semantic role labels as a function of the gold-standard part of speech of the argument, the gold-standard lemma of the predicate and the flag whether there is a syntactic dependency between the two nodes or not. This approach makes it theoretically possible to make one token semantically dependent on more than one predicate. However, we have no means to control the number of the dependencies.

3 Results

The official results of our system are given in Table 1. The system made the least syntactic errors (attachment and labels) for Japanese. The Japanese treebank seems to be relatively easy to parse, as many other systems achieved very high scores on this data. At the other end of the rating scale, Chinese seems to be the syntactically hardest language. Our second-worst syntactic score was for Czech, most likely owing to the turning off all language-dependent (and Czech-biased) features of the parser.

An obvious feature of the table is the extremely poor semantic scores (in contrast to the accuracy of surface dependency attachment and labels). While the simplicity of the additional models does not seem to hurt too much the dependency labeling, it apparently is too primitive for semantic role labeling. We analyze the errors in more detail in Section 4.

The system is platform-independent;[3] we have been running all the experiments under Linux on an AMD Opteron 848 processor, 2 GHz, with 32 GB RAM. The running times and memory requirements are shown in Table 2.

[2] However, the parser – without adaptation – would not do well on Czech anyway because the PDT tags are presented in a different format in the shared task data.

[3] It is written entirely in Perl.

Language	Average	Cs	En	De	Es	Ca	Ja	Zh
Labeled macro F1	51.07	**57.95**	50.27	49.57	48.90	49.61	57.69	43.50
OOD lab mac F1	43.67	**54.49**	48.56	27.97				
Labeled syn accur	64.92	57.06	61.82	69.79	65.98	67.68	**82.66**	49.48
Unlab syn accur	70.84	66.04	70.68	72.91	71.22	73.81	**83.36**	57.87
Syn labeling accur	79.20	69.10	74.24	84.63	81.83	82.46	**95.98**	66.13
OOD lab syn acc	50.20	51.45	**62.83**	36.31				
OOD unl syn acc	58.08	60.56	**71.78**	41.90				
OOD syn labeling	69.65	65.64	**75.22**	68.08				
Semantic lab F1	32.14	**58.13**	36.05	16.44	25.36	24.19	30.13	34.71
OOD sem lab F1	32.86	**56.83**	31.77	9.98				

Table 1. The official results of the system. ISO 639-1 language codes are used (cs = Czech, en = English, de = German, es = Spanish, ca = Catalan, ja = Japanese, zh = Chinese). "OOD" means "out-of-domain test data".

Language	Cs	En	De	Es	Ca	Ja	Zh
Training sentences	43955	40613	38020	15984	14924	4643	24039
Training tokens	740532	991535	680710	477810	443317	119144	658680
Average sentence length	17	24	18	30	30	26	27
Training minutes	9:21	10:41	8:28	6:17	5:42	1:24	7:01
Training sentences per secnd	78	63	75	42	44	55	57
Training tokens per second	1320	1547	1340	1267	1296	1418	1565
Training rsize memory	3.9 GB	2.2 GB	2.7 GB	2.7 GB	2.4 GB	416 MB	1.5 GB
Test sentences	4213	2399	2000	1725	1862	500	2556
Test tokens	70348	57676	31622	50630	53355	13615	73153
Parsing minutes	6:36	3:11	2:24	5:47	6:05	0:46	5:45
Parsing sentences per second	10.6	12.6	13.9	5.0	5.1	10.9	7.4
Parsing tokens per second	178	302	220	146	146	296	212
Parsing rsize memory	980 MB	566 MB	779 MB	585 MB	487 MB	121 MB	444 MB

Table 2. Time and space requirements of the syntactic parser.

To assess the need for data, Table 3 presents selected points on the learning curve of our system. The system has been retrained on 25, 50 and 75% of the training data for each language (the selection process was simple: the first N% of sentences of the training data set were used).

Generally, our method does not seem very data-hungry. Even for Japanese, with the smallest training data set, reducing training data to 25% of the original size makes the scores drop less than 1% point. The drop for other languages lies mostly between 1 and 2 points. The exceptions are (unlabeled) syntactic attachment accuracies of Czech and Spanish, and labeled semantic F1 of Spanish and Chinese. The Chinese learning curve also contains a nonmonotonic anomaly of syntactic dependency labeling between data sizes of 50 and 75% (shown in boldface). This can be probably explained by uneven distribution of the labels in training data.

As to the comparison of the various languages and corpora, Japanese seems to be the most specific (relatively high scores even with such small data). Spanish and Catalan are related languages, their treebanks are of similar size, conform to similar guidelines and were prepared by the same team. Their scores are very similar.

4 Discussion

In order to estimate sources of errors, we are now going to provide some analysis of the data and the errors our system does.

4.1 DEPREL Coverage

The syntactic tagger (assigns DEPREL syntactic labels) and the semantic tagger (assigns PRED and APRED labels) are based on simple statistical models without sophisticated back-off techniques.

Score	TrSize	Average	Cs	En	De	Es	Ca	Ja	Zh
UnLab Syn Attach	**25%**	69.38	63.72	69.70	71.36	68.99	72.41	82.58	56.90
	50%	70.14	64.96	70.13	72.11	70.37	72.83	82.99	57.58
	75%	70.51	65.50	70.37	72.50	70.83	73.47	83.17	57.73
	100%	70.84	66.04	70.68	72.91	71.22	73.81	83.36	57.87
Syn Label	**25%**	78.47	68.28	73.79	84.21	80.67	81.92	95.70	64.71
	50%	78.94	68.68	74.08	84.44	81.59	81.99	95.86	**65.94**
	75%	79.03	68.87	74.14	84.51	81.67	82.19	95.97	**65.83**
	100%	79.20	69.10	74.24	84.63	81.83	82.46	95.98	66.13
Labeled Sem F1	**25%**	30.10	56.29	34.47	15.51	22.78	22.14	28.91	30.58
	50%	**33.85**	57.24	35.34	16.03	24.46	23.13	29.60	33.31
	75%	**31.76**	57.76	35.85	16.29	24.96	23.77	29.96	33.71
	100%	32.14	58.13	36.05	16.44	25.36	24.19	30.13	34.71
Labeled Macro F1	**25%**	49.19	55.87	49.06	48.10	46.22	47.76	56.66	40.64
	50%	50.28	56.99	49.66	48.90	47.97	48.53	57.23	42.66
	75%	50.68	57.53	50.01	49.26	48.47	49.21	57.52	42.73
	100%	51.07	57.95	50.27	49.57	48.90	49.61	57.69	43.50

Table 3. The learning curve of the principal scores.

Sparse data could pose a serious problem. So how sparse are the data? Some cue could be drawn from Table 3. However, we should also know how often the labels had to be assigned to an unknown set of input features.

DEPREL (syntactic dependency label) is estimated based on morphological tag (i.e. POS + FEAT) of both the child and parent. If the pair of tags is unknown, then it is based on the tag of the child, and if it is unknown, too, the most frequent label is chosen. Coverage is high: 93 (Czech) to 97 % (Chinese) of the pairs of tags in test data were known from training data. Moreover, the error rate on the unknown pairs is actually much *lower* than on the whole data![4]

4.2 PRED Coverage

PRED (predicate sense label) is estimated based on lemma. For most languages, this seems to be a good selection. Japanese predicate labels are always identical to lemmas; elsewhere, there are by average 1.05 (Chinese) to 1.48 (Spanish) labels per lemma; the exception is German with a label-lemma ratio of 2.33.

Our accuracy of PRED label assignment ranges from 71% (German) to 100% (Japanese). We always assign the most probable label for the given lemma; if the lemma is unknown, we copy the lemma to the PRED column. Coverage is not an issue here. It goes from 94% (Czech) to almost 100% (German).[5] The accuracy on unknown lemmas could probably be improved using the subcategorization dictionaries accompanying the training data.

Language	Lemma	PREDs
Cs	1. mít	77
	2. přijmout	8
En	1. take	20
	2. go	18
De	1. kommen	28
	2. nehmen	25
Es	1. pasar	10
	1. dar	10
	3. llevar	9
	3. hacer	9
Ca	1. fer	11
	2. pasar	9
Ja	*Always 1 PRED per lemma*	
Zh	1. 要 (yào)	8
	1. 有 (yǒu)	8
	1. 打 (dǎ)	8

Table 4. Most homonymous predicates.

[4] This might suggest that the input features are chosen inappropriately and that the DEPREL label should be based just on the morphology of the child.

[5] The coverage of Japanese is 88% but since Japanese PRED labels are exact copies of lemmas, even unknown lemmas yield 100%-correct labels.

Language	Cs	En	De	Es	Ca	Ja	Zh
Potential APRED slots	1287545	195029	12066	192103	197976	57394	329757
Filled in APREDs	87934	32968	10480	49904	52786	6547	49047
Feature pair coverage (%)	46.05	40.04	14.99	29.34	29.89	18.31	38.08
Non-empty APRED accuracy	73.19	64.65	67.37	56.90	57.89	59.20	68.77
Unlabeled precision	34.94	26.86	10.88	21.71	20.25	9.13	25.66
Unlabeled recall	62.61	63.86	97.52	93.40	92.72	22.10	67.82
Unlabeled F	44.86	37.81	19.57	35.23	33.24	12.93	37.23
Labeled precision	25.58	17.36	7.33	12.35	11.72	5.41	17.64
Labeled recall	45.83	41.28	65.70	53.15	53.67	13.08	46.64
Labeled F	32.83	24.44	13.19	20.05	19.24	7.65	25.60

Table 5. APRED detailed analysis. Non-empty APRED accuracy includes only APRED cells that were non-empty both in gold standard and system output. Feature-pair coverage includes all cells filled by the system. Unlabeled precision and recall count non-empty vs. empty APREDs without respect to their actual labels. Counted on development data with gold-standard surface syntax.

4.3 APRED Assignment Analysis

The most complicated part of the task is the assignment of the APRED labels. In a sense, APRED labeling is dependency parsing on a deeper level. It consists of several sub-problems:

- Is the node an argument of any predicate at all?
- If so, how many predicates is the node argument of? Should the predicate be, say, coordination, then the node would semantically depend on all members of the coordination.
- In what way is the semantic dependency related to the syntactic dependency between the node and its syntactic parent? In majority of cases, syntactic and semantic dependencies go parallel; however, there are still a significant number of semantic relations for which this assumption does not hold.[6]
- Once we know that there is a semantic relation (an APRED field should not be empty), we still have to figure out the correct APRED label. This is the semantic role labeling (or tagging) proper.

Our system always makes semantic roles parallel to surface syntax. It even does not allow for empty APRED if there is a syntactic dependency—this turned out to be one of the major sources of errors.[7]

The role labels are estimated based on the lemma of the predicate and the part of speech of the argument. Low coverage of this pair of features in the training data turns to be another major source of errors. If the pair is not known from training data, the system selects the most frequent APRED in the given treebank. Table 5 gives an overview of the principal statistics relevant to the analysis of APRED errors.

5 Post-evaluation Experiments

Finally, we performed some preliminary experiments focused on the syntactic parser. As mentioned in Section 2.1, many features of the parser have to be turned off unless the parser understands the part-of-speech and morphological features. We used DZ Interset (Zeman, 2008) to convert Czech and English CoNLL POS+FEAT strings to PDT-like positional tags. Then we switched back on the parser options that use up the tags and re-ran parsing. The results (Table 6) confirm that the tag manipulation significantly improves Czech parsing while it does not help with English.

[6] Nearly all Spanish and Catalan semantic dependencies are parallel to syntactic ones (but not all syntactic dependencies are also semantic); in most other languages, about two thirds of semantic relations match syntax. Japanese is the only language in which this behavior does not prevail.

[7] This is a design flaw that we overlooked. Most likely, making empty APRED one of the predictable values would improve accuracy.

	Cs	En
Before	65.81	69.48
After	71.76	68.92

Table 6. Unlabeled attachment accuracy on development data before and after tagset conversion.

6 Conclusion

We described one of the systems that participated in the CoNLL 2009 Shared Task. We analyzed the weaknesses of the system and identified possible room for improvement. The most important point to focus on in future work is specifying where APRED should be filled in. The links between syntactic and semantic structures have to be studied further. Subcategorization frames could probably help improve these decisions, too—our present system ignores the subcategorization dictionaries that accompany the participating treebanks.

Acknowledgments

This research has been supported by the Ministry of Education of the Czech Republic, project No. MSM0021620838.

References

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó and Manfred Pinkal. 2006. The SALSA Corpus: a German Corpus Resource for Lexical Semantics. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006).* Genova, Italy.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antonia 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. *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009).* June 4-5. pp. 3-22. Boulder, Colorado, USA.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová and Zdeněk Žabokrtský. 2006. *The Prague Dependency Treebank 2.0.* CD-ROM. Linguistic Data Consortium, Philadelphia, Pennsylvania, USA. ISBN 1-58563-370-4. LDC Cat. No. LDC2006T01. URL: http://ldc.upenn.edu/.

Daisuke Kawahara, Sadao Kurohashi and Koiti Hasida. 2002. Construction of a Japanese Relevance-tagged Corpus. *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002).* pp. 2008-2013. Las Palmas, Spain.

Mihai Surdeanu, 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 the 12th Conference on Computational Natural Language Learning (CoNLL-2008).* August 16 – 17. Manchester, UK.

Mariona Taulé, Maria Antònia Martí and Marta Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-2008).* Marrakech, Morocco.

Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the Chinese Treebank. *Natural Language Engineering,* **15**(1):143-172.

Daniel Zeman. 2004. *Parsing with a Statistical Dependency Model* (PhD thesis). Univerzita Karlova, Praha, Czechia. URL: http://ufal.mff.cuni.cz/~zeman/projekty/parser/index.html

Daniel Zeman. 2008. Reusable Tagset Conversion Using Tagset Drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-2008).* ISBN 2-9517408-4-0. Marrakech, Morocco.