

# Riga: from FrameNet to Semantic Frames with C6.0 Rules

Guntis Barzdins, Peteris Paikens, Didzis Gosko

University of Latvia, IMCS

Rainis Blvd. 29, Riga, LV-1459, Latvia

{guntis.barzdins,peteris.paikens,didzis.gosko}@lumii.lv

## Abstract

For the purposes of SemEval-2015 Task-18 on the semantic dependency parsing we combined the best-performing closed track approach from the SemEval-2014 competition with state-of-the-art techniques for FrameNet semantic parsing. In the closed track our system ranked third for the semantic graph accuracy and first for exact labeled match of complete semantic graphs. These results can be attributed to the high accuracy of the C6.0 rule-based sense labeler adapted from the FrameNet parser. To handle large SemEval training data the C6.0 algorithm was extended to provide multi-class classification and to use fast greedy search without significant accuracy loss compared to exhaustive search. A method for improved FrameNet parsing using semantic graphs is proposed.

## 1 Introduction

The trend of natural language processing in recent years is shifting towards multilingual natural language understanding based on full-text shallow semantic parsing (e.g., Banarescu et al., 2013). Despite various formalisms proposed, these approaches are characterized by direct extraction of a bi-lexical semantic graph rather than a bi-lexical dependency tree from the surface form of the sentence.

Following the best practice for semantic parsing established already by the SemEval-2014 Task 8 (Oepen et al., 2014) we modified the best-performing closed-track system there (Du et al., 2014) by removing some less essential components

while adding a new component of our own. The newly added component is the C6.0 rule-based classifier (Barzdins et al., 2014) used both for graph parsing and for sense labeling. Sense labeling is a novelty of SemEval-2015 Task 18 and was not present in the previous year competition. Semantic frame is comprised of a complete predication combined with the sense identifier of its predicate as shown in Figure 1. Semantic frames are similar to FrameNet (Fillmore et al., 2003) frames, except that FrameNet argument labels are sense-specific – this mismatch can be resolved by feeding the semantic graph (instead of dependency tree) through the regular FrameNet parser.

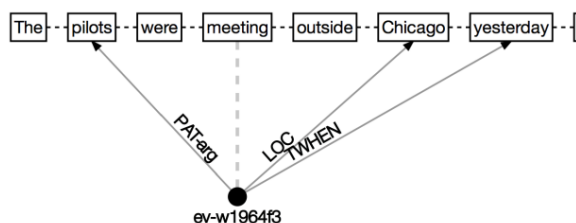


Figure 1. Semantic frame from the PSD corpus.

We participated only in the closed track. Despite ranking third for the semantic graph accuracy, our system ranked first for exact labeled match of complete semantic graphs, and close second for semantic frame accuracy.

## 2 Baseline Architecture

For semantic graph parsing we started by implementing a straight-forward baseline architecture described on the SemEval-2015 Task-18 evaluation page by the task organizers. The baseline architecture consists of two components: reduction

of the SDP graphs to trees and training the Mate-tools dependency parser (Bohnet, 2010) to produce such trees from the unparsed text. Instead of a destructive reduction of the SDP graphs to trees, we implemented a fully reversible depth-first transformation from the last year best-performing system (Du et al., 2014). This simple approach immediately produced competitive graph parsing results (Table 1) in line with the best-performing systems from the last year.

	in domain			out of domain		
	LP	LR	LF	LP	LR	LF
en.dm	87.34	87.05	87.19	79.95	79.42	79.68
en.pas	90.47	90.03	90.25	85.98	85.48	85.73
en.psd	72.81	71.05	71.92	70.34	67.55	68.92
cs.psd	74.44	71.56	72.97	60.19	57.43	58.78
cz.pas	82.15	81.74	81.94	-	-	-

Table 1. Baseline architecture labeled scores.

For sense labeling in *en.dm* and *en.psd* representations (a new task not present in the previous SemEval-2014 competition) we reused a technique from prior work on FrameNet labeling (Barzdins et al., 2014) based on C6.0 classifier<sup>1</sup>. For this task the C6.0 classifier was modified (see Section 3) to directly produce the multi-class output. By using as the features values from the *form*, *lemma*, *POS* columns for the previous, current, and next word, this approach gave good results on the development set: 93.86% accuracy for *en.psd* representation and 94.50% accuracy for *en.dm* representation. We did not try to improve it any further and the same baseline approach was used also for producing senses in the final submitted parses.

In the submitted parses we carried out the graph parsing and sense labeling completely independently, naively combining both annotations afterwards. Later experiments have shown that using graph parsing results as additional features for sense labeling would improve sense accuracy by approximately 0.2%.

### 3 Sense Labeling with C6.0 Rules

C6.0 rule-based classification algorithm (Barzdins et al., 2014) was inspired by the popular C4.5 decision-tree classification algorithm (Quinlan, 1993)

<sup>1</sup> Available at <http://c60.aillab.lv>

and has been used in the state-of-the-art FrameNet parser.

To accommodate the large training data sets provided in SemEval competition we extended the original C6.0 algorithm with support for the multi-class classification and with the fast greedy search as a replacement for the exhaustive search in the original C6.0 version.

Given  $k$  training examples of the form:

$$\begin{aligned}
 & (a_{11}, a_{12}, a_{13}, \dots a_{1n}, \text{class}_1) \\
 & (a_{21}, a_{22}, a_{23}, \dots a_{2n}, \text{class}_2) \\
 & \dots \\
 & (a_{k1}, a_{k2}, a_{k3}, \dots a_{kn}, \text{class}_k)
 \end{aligned}$$

where features  $a_{ij}$  and  $\text{class}_i$  are arbitrary character strings, C6.0 classifier builds a list of rules (illustrated in Figure 2) for predicting the class of unseen examples. The left side of the rule is a pattern where any feature position may contain a specific character string to be matched or an unspecified value denoted by “\_”.

	lemma	POS		Predicted sense	$p$	$n$	Laplace ratio
<i>iff</i>	the,	DT	)then	q:i-h-h	227	0	0.996
<i>iff</i>	,	CD	)then	card:i-i-c	147	9	0.937
<i>iff</i>	.	DT	)then	q:i-h-h	336	31	0.913
<i>iff</i>	trade,		)then	n.of:x-i	13	1	0.875

Figure 2. Classification rules generated by C6.0. Rule quality is estimated by the Laplace ratio based on positive  $p$  and negative  $n$  matching training examples.

The greedy search algorithm for building a multi-class classifier can be described as follows.

Training data is converted to a pool of classifier training examples. Each training example is considered positive for the class it belongs to, and negative for any other class. A candidate rule is matched against all positive and negative training examples relative to its class. The count of matched positive and negative training examples allows to calculate rule’s Laplace ratio  $(p+1)/(p+n+2)$ , where  $p$  is the number of matching positive training examples and  $n$  is the number of matching negative training examples. The rules with higher Laplace ratio are better.

For each training example a set of rules correctly classifying this training example is generated by incrementally adding to the left side of the rule feature values from this training example. Fast greedy search one-by-one adds the features in such

order that the resulting rule has the highest possible Laplace ratio in every feature adding iteration. This is contrary to the original C6.0 exhaustive search strategy which tried all feature relaxation combinations instead. The greedy approach eliminates exponential complexity of C6.0 with respect to feature count and when tested, yielded as good results as the exhaustive search on SemEval data.

All generated rules (regardless of the class they predict) are sorted by the highest Laplace ratio. The resulting list of rules is a multi-class classifier which can be considered consisting of multiple binary classifiers (individual rules). For unseen examples the class is assigned by the matching rule with the highest Laplace ratio.

Fig. 2 shows some classification rules for predicting the sense column value in *en.dm* training dataset from two features. The actual production classifier for sense labeling uses more features (listed in Section 2) and generates several thousand rules.

## 4 Semantic Graph Parsing

We tried three approaches described below to improve the graph parsing results above the baseline.

### 4.1 Peking and MateTools Graph Parser

The primary approach chosen for semantic graph parsing is to implement a fully reversible transformation between the semantic graph and a tree representation that encodes the extra information in edge labels. It allows training a dependency parser (Bohnet, 2010) on the labeled tree data, and using it to parse text to structures that can be converted back to a semantic graph.

For reversible graph to tree transformation we have implemented the depth-first search transformation and the auxiliary label system used by last year’s best-performing Peking system (Du et al, 2014). The auxiliary labels encode:

- A separator to indicate multiple original edges encoded in this label;
- Ancestor-number indicating that in the original graph, an edge with this label is drawn from the dependent to the n-th ancestor instead of the direct parent of this tree edge;

- A reverse-edge symbol to indicate edges that have reversed direction compared to the original graph.

For the multi-root sentences that appear in some of the datasets, we choose the first root (according to word order in sentence) as the main tree root, and iteratively link all the other sentence fragments to the nearest node in the accumulated tree according to the number of words between them; in case of ties preferring the leftmost node. When creating the transformed tree, we also used special labels to distinguish the secondary root nodes of other fragments, so that the transformation is reversible for graphs with multiple root nodes.

After parsing, a tree may contain labels that are invalid according to the principles of this transformation – i.e., a reference to the grandparent of a node that does not have one. In this case, we draw an edge with the appropriate label to the closest possible node.

In this approach the cyclic graph structures are transformed to the different tree branch topologies depending on the traversal order. Traversal order thus affects the likelihood of the parser to correctly reconstruct these cyclic graph structures. To improve cyclic graph structure reconstruction we developed multiple parser variations for ensemble voting based on the following traversal orders for each node:

- Linear distance of linked words, starting with the closest words and preferring the left node in case of ties;
- Frequency of the edge labels, prioritizing the most frequent labels;

In addition, we also applied the same transformations for sentences with reversed word order to provide further variation. The resulting parsers have comparable accuracy, but produce different mistakes, making them useful for ensemble voting. Simple ensemble voting improves graph parsing accuracy over the baseline (Table 2).

	in domain			out of domain		
	LP	LR	LF	LP	LR	LF
en.dm	88.63	87.12	87.87	81.75	79.61	80.67
en.pas	91.46	90.01	90.73	87.55	85.71	86.62
en.psd	75.25	71.29	73.22	73.28	67.52	70.28
cs.psd	78.66	71.73	75.04	64.27	57.72	60.82
cz.pas	83.10	81.85	82.47	-	-	-

Table 2. Ensemble method labeled scores.

## 4.2 C6.0 Rule Based Graph Parser

We also applied our C6.0 rule-based classifier (described in Section 3) for semantic graph parsing through exact dependency phrase matching. Due to low recall rate it provided only a tiny positive boost to the final ensemble voting result (Table 4) despite the high precision of the rules method (Table 3). Here we considered only edges of length up to 4 and C6.0 rules with Laplace ratio above 90%. Due to low recall we signaled “abstain” vote for the edges not covered by these rules.

	in domain			out of domain		
	LP	LR	LF	LP	LR	LF
en.dm	92.80	33.47	49.20	91.84	19.78	32.56
en.pas	92.94	35.53	51.40	92.58	28.07	43.08
en.psd	88.34	18.76	30.94	86.70	11.34	20.05
cs.psd	95.29	16.70	28.42	80.46	8.13	14.77
cz.pas	90.97	22.91	36.60	-	-	-

Table 3. Labeled scores for the rules method.

## 4.3 Other parsing approaches

Experiments with transition based parsers (Malt-Parser/MaltOptimizer) showed approximately 2% lower accuracy than Mate-tools on the same transformed tree data. This is consistent with findings made by others during the earlier SemEval-2014 Task-8. We chose not to use those parsers for the final submission.

## 5 Final Results

We submitted two runs but report results only for run-1, because run-2 was discovered to include a corrupted Mate-tools dataset.

Our final semantic graph and semantic frames parsing results are shown in Tables 4 and 5. Semantic frames results measure overall sense labeling and graph parsing accuracy, which is the novelty of this year SemEval task.

	in domain			out of domain		
	LP	LR	LF	LP	LR	LF
en.dm	88.57	87.24	87.90	81.69	79.72	80.69
en.pas	91.50	90.02	90.75	87.56	85.72	86.63
en.psd	75.25	71.52	73.34	73.23	67.71	70.37
cs.psd	78.66	71.84	75.10	64.29	57.83	60.89
cz.pas	83.12	81.84	82.47	-	-	-

Table 4. Labeled scores for the submitted result.

	in domain			out of domain		
	FP	FR	FF	FP	FR	FF
en.dm	58.45	57.79	58.12	42.62	41.17	41.88
en.psd	52.48	52.59	52.54	40.60	40.93	40.76

Table 5. Semantic frame scores for the submitted result.

Table 6 shows ranking of averaged SemEval scoring metrics for the best runs of the systems participating in the closed task. Although we ranked third for the semantic graph (labeled dependencies) metric, our system ranked close second for semantic frame accuracy, and first for labeled exact match of the complete semantic dependency graphs. These results suggest that the C6.0 rule accuracy for sense labeling and for exact match semantic graph parsing was able to compensate for slightly lower overall graph parsing accuracy.

System	LF	LM	PF	SF	FF
Peking	<b>80.51</b>	21.14	62.64	69.45	<b>48.70</b>
Lisbon	80.42	20.05	<b>63.59</b>	--	--
Riga	78.68	<b>21.84</b>	61.29	73.76	48.33
Minsk	78.18	15.04	56.40	<b>79.40</b>	47.32

Table 6. Ranking of scores averaged over all available datasets for the best runs of the systems in the closed track: labeled dependencies (LF), labeled exact match of the complete semantic dependency graphs (LM), complete predications (PF), sense identification (SF), semantic-frames (FF).

## 6 Conclusions

Variations of Peking depth-first reversible graph-to-tree conversion algorithm in combination with state-of-the-art dependency parser is still a competitive graph parsing approach.

C6.0 rule-based classifier provides competitive sense labeling accuracy and some improvement also for graph parsing accuracy.

An ensemble method with “abstain” voting option for joining outputs of various graph parsing approaches boosts the results by ironing out the weaknesses of individual parsers. Required computational resources are the main limitation here.

## Acknowledgments

This work was supported by the Latvian National research program SOPHIS under grant agreement Nr.10-4/VPP-4/11. We thank Lauma Pretkalniņa for the experiments with transition-based parsers.

## References

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In: *Proc. Linguistic Annotation Workshop (SIGANN-2013)*, pp. 178-186.
- Guntis Barzdins, Didzis Gosko, Laura Rituma, and Peteris Paikens. 2014. Using C5.0 and Exhaustive Search for Boosting Frame-Semantic Parsing Accuracy. In: *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*, pp. 4476-4482.
- Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is not a Contradiction. *The 23rd International Conference on Computational Linguistics (COLING 2010)*, pp. 89-97.
- Charles J. Fillmore, Christopher R. Johnson, and Miriam R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16, pp. 235-250.
- John R. Quinlan. 1993. C4.5: Programs for Machine Learning. *Morgan Kaufmann Publishers*. 302 p.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pp. 63-72.
- Yantao Du, Fan Zhang, Weiwei Sun, and Xiaojun Wan. 2014. Peking: Profiling Syntactic Tree Parsing Techniques for Semantic Graph Parsing. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pp. 459-464.