Dependency and Phrasal Parsers of the Czech Language: A Comparison

Aleš Horák[1], Tomáš Holan[2], Vladimír Kadlec[1], Vojtěch Kovář[1]

[1] Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{hales,xkadlec,xkovar3}@fi.muni.cz
[2] Faculty of Mathematics and Physics
Charles University
Malostranské nám. 25, CZ-11800 Prague, Czech Republic
Tomas.Holan@mff.cuni.cz

Abstract. In the paper, we present the results of an experiment with comparing the effectiveness of real text parsers of Czech language based on completely different approaches – stochastic parsers that provide dependency trees as their outputs and a meta-grammar parser that generates a resulting chart structure representing a packed forest of phrasal derivation trees.
We describe and formulate main questions and problems accompanying such experiment, try to offer answers to these questions and finally display also factual results of the tests measured on 10 thousand Czech sentences.

1 Introduction

During last ten years a number of syntax parsers of the Czech language have been implemented with the concentration to real parsing of real texts (in contrast to theoretical and demonstration parsers created in 80s and 90s of the last century).

Some of those "real text parsers" came into existence in the team around the Prague Dependency Treebank [1], we will call them as the Prague parsers although the best ones of them are variants of parsers of British or American authors.

The other set of compared parsers are variants of the parser designed and implemented in the team of NLP laboratory at Masaryk university in Brno (the `synt` parser [2]), thus we call it the Brno parser in the context of this paper.

Although all these parsers are tested and used for several years already, their implementations are running more or less independently and no rigorous comparison of their effectiveness has been done yet.

This paper tries to formulate all problems that have hindered such comparison so far, then offers a solution of them and finally present the results of the actual comparison. The Prague parsers have already been compared and rated all together, so the novelty in this comparison is the Brno parser `synt` that is based on completely different approaches than the Prague parsers.

[3] This work has been partly supported by the Academy of Sciences of Czech Republic under the projects T100300414, T100300419 and 1ET100300517 and by the Ministry of Education of CR within the Center of basic research LC536 and by the Czech Science Foundation under the project 201/05/2781.

2 The Compared Parsers

In this section, we will shortly describe the parsers used in the prepared measuring and comparison.

2.1 The Prague Parsers – Basic Characteristics

The set of dependency parsers selected and denoted as the Prague parsers contains the following representatives:

McD McDonnald's maximum spanning tree parser [3],
COL Collins's parser adapted for PDT [4],
ZZ Žabokrtský's rule-based dependency parser [5],
AN Holan's parser ANALOG – it has no training phase and in the parsing phase it searches in the training data for the most similar local tree configuration [6],
L2R, R2L, L2R3, R2L3 Holan's pushdown parsers [7],
CP Holan's and Žabokrtský 's combining parser [5],

The selection of Prague parsers was limited to the parsers contained in CP, which is currently the parser with the best known results on PDT including also other parsers like, e.g., Hall and Novák's corrective modeling parser [8] or Nilsson, Nivre and Hall's graph transformation parser [9]. These parsers were not included in the comparison, since currently we do not have their results for all sentences of the testing data set.

The pushdown parsers, during their training phase, create a set of premise-action rules, and apply it during the parsing phase. In the training phase, the parser determines the sequence of actions which leads to the correct tree for each sentence (in case of ambiguity, a pre-specified preference ordering of the actions is used). During the parsing phase, in each situation the parser chooses the premise-action pair with the highest score. In the tests, we have measured four versions of the pushdown parser: L2R – the basic pushdown parser (left to right), R2L – the parser processing the sentences in reverse order, L23 and R23 – the parsers using 3-letter suffices of the word forms instead of the morphological tags.

2.2 The Brno Parser – Basic Characteristics

In contrast to the Prague parsers, the Brno parser `synt` is based only on its meta-grammar, the parser does not have any training phase used to learn the context dependencies of the input texts. All rules that guide the analysis process are developed by linguistic and computer experts with all the drawbacks it can bring (see the Section 3.5 for a description of some of them). The advantage of this process is a better adaptation to yet undescribed language phenomena.

The current meta-grammar contains about 250 meta-rules that allow to describe in a human-maintainable way all possible rules used as the actual input for the chart parsing algorithm formed by 2800 generated rules plus feature agreement tests and other contextual actions used for pruning the resulting chart. This meta-grammar describes more than 90 % of sentences from the PDTB-1.0 corpus (the predecessor of PDT-2.0).

The involved chart parsing algorithm uses a modified *Head-driven chart parser* [10], which provides a very fast parsing of real-text sentences with an average time of 0.07sec/sentence.

3 The Principal Differences of the Parsers

The most principal difference between the parsers is, of course, the underlying formalism and methodology of the parsing process. This is however not the sort of difference that would cause problems in the parser comparison. In this section, we will concentrate on the problems arising with different input and output data structures, different morphological and syntactical tagging and different presuppositions on the input text that all need to be resolved before we can start with the real comparison.

3.1 Q1: The Input Format

The input of the Brno parser is either a tagged text (from corpus or from other tagged source) with morphological tags compatible with the tagset of the Czech morphological analyser called Ajka [11] or a plain text (divided into sentences), which is then processed with Ajka. Since Ajka does not resolve ambiguities on the morphological level,[3] the Brno parser generally counts with the possibility of ambiguous surface level tokens.

The Prague parsers use as their input also text split into individual sentences, but with unambiguous morphological tags obtained from Hajič's morphological analyser and tagger [12].

Both morphological analysers (and thus both parser groups) use different morphological tagging systems, which are not 1:1 translatable to each other. However, the differences do not affect the most important morphological features from the point of view of the syntactic analysis, so we have used an automatic conversion with some information stripping.

3.2 Q2: Dependency Trees vs. Phrasal Trees

The output of Prague parsers is formed by dependency trees or graphs, whereas the output of the Brno parser is basically formed by packed shared forest of phrasal trees. The Brno parser includes the possibility of sorting the trees of the shared forest and output N trees with the highest *tree rank* (a value obtained as a combination of several "figures of merit," see [13]).

This difference in the output format plus the fact that the Brno team does not yet have a large testing tree bank of phrasal trees for measurements[4] was the cause of the biggest problems in the comparison. Since the measurements had to be done on several thousands of sentences, we have decided to use the PDT-2.0 tree bank[5] [14]. Since this tree bank provides only the dependency trees for more than 80 thousand Czech sentences, we have decided to convert them to phrasal trees using the Collin's conversion tool [15] and then measure the differences between the Brno parser output and this "phrasal PDT-2.0" using the *PARSEVAL* and the *Leaf-ancestor assessment* techniques (for more details see the Section 4).

[3] Ajka provides all possible combinations of morphological features of the input words.

[4] Such tree bank of about 5 thousand phrasal trees is being prepared during this year.

[5] The Prague Dependency Treebank, version 2.0, was created by the Institute of Formal and Applied Linguistics, `http://ufal.mff.cuni.cz`.

3.3 Q3: One Resulting Tree vs. (Shared) Forest

The output of the Brno parser is formed by the resulting *chart* structure, which encompasses a whole forest of derivation trees (all of them, however, have the same root nonterminal that represents the successful analysis).

In order to be able to provide a comparison of this forest with the one tree obtained from PDT 2.0 conversion procedure, we have for each sentence extracted first 100 (or less) trees sorted according to the *tree rank*. Each of these trees was then compared to the one from PDT and the results are displayed with the following 3 numbers: a) *best trees* – one tree from the set that is most similar to the desired tree is selected and compared; b) *first tree* – the tree with the highest tree rank is selected and compared; and c) *average* – the average of all trees is presented.

3.4 Q4: Projective vs. Non-projective Trees

The output of the Brno parser is always in the form of projective trees, but a non-projective phrase can, in some cases, be analysed with the mechanism of different rule levels, that allow to handle special kinds of phrases. Nevertheless, the Brno parser is not suitable for analysing non-projective sentences at the moment. In the future, we will have to provide techniques like corrections for non-projective parses described in [8].

On the other hand, the output of the Prague parsers, as a set of dependency edges between words, can cross the word surface order without problems. Thus it can represent projective as well as non-projective sentences.

According to the Prague Dependency Treebank statistics, PDT contains approximately 20 % of non-projective sentences. The sentences selected for comparison are thus not limited to only projective sentences, but the results are counted separately for projective and non-projective sentences.

3.5 Q5: The Testing Data Set

For the measuring and comparison of parser effectiveness, we definitely need syntactically annotated data. Such data are available for the dependency trees in PDT. The tree bank has three parts – the training part (train), the testing part for development (d-test) and the testing part for evaluation (e-test).

Since the Prague parsers use the first two sets for development and because there is no such similar tree bank available for the phrasal trees from the Brno parser, we have decided to use the PDT e-test part (approx. 10 thousand sentences) for the comparison and we will try to overcome the differences between the parser outputs.

One important difference regarding the testing data set is the fact that the Brno parser does not have any training or learning phase – it is purely grammar based parser. The drawback of this fact is that the Brno parser cannot automatically adapt to kinds of texts that were not intended for analysis. The parser is designed to analyse only sentences of the usual structure. Since the Czech language is a representative of free-word-order languages, the parser allows an analysis of many possible word combinations that can form even very “wild” Czech sentences, however, it refuses to analyse texts like PDT sentences e-test#00017: “*10 - 3 %*” or e-test#00554: “*Dítě 4 - 10 let : 1640* (Child 4–10 years:1640).” The Prague parsers, thanks to their stochastic nature, do not have any problems in analysing such kinds of sentences.

4 The Results

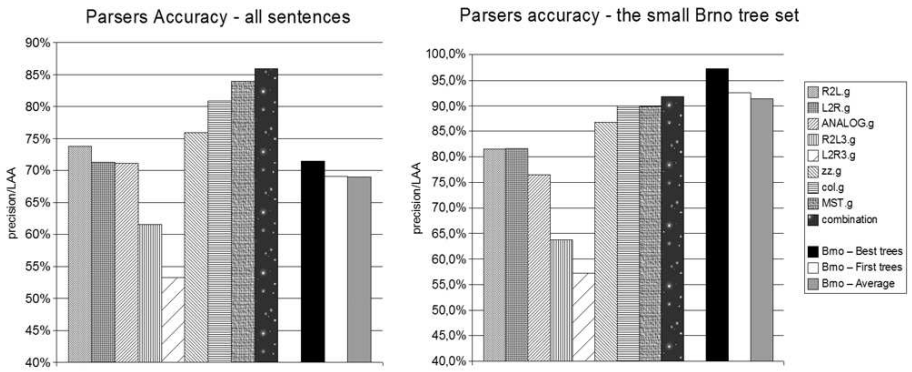

Fig. 1. The difference of the results with measuring on the converted PDT trees and on the small Brno tree set

As we have described in the Section 3, we have decided to use the PDT-2.0 e-test part, where the morphological tags were automatically converted from the Prague tags to the Ajka tags without ambiguities. The e-test set contains approximately 10 thousand syntactically annotated dependency trees. To get trees comparable to Brno parser output, we needed to convert these dependency trees to phrasal trees.

The conversion proceeded in two steps: first, the PDT-2.0 dependency trees in PML format (the default format in PDT-2.0) were converted into the CSTS format (earlier format of PDT) with PDT tool `btred`. Then, the Collin's conversion tool [15] was used to obtain PDT-2.0 phrasal trees similar to the output of the Brno parser. The statistical features of the e-test set are:

- 10148 sentences (173586 words)
- 7732 projective sentences
- 2416 non-projective sentences
- 87.7 % Brno parser coverage

Since the Brno parser does not provide output for all sentences in the e-test set (see the discussion in the Section 3.5), the actual comparison was run only on those sentences from e-test, that were accepted by the Brno parser.

4.1 Measuring Techniques

The methodology for measuring the results of dependency parsing is usually defined as computation of the precision and recall of the particular dependency edges in the resulting graph/tree. These quantities are measured for each lexical item and the result is then computed as an average precision and average recall throughout the whole set.

Parser	all sentences	non-projective	projective
R2L	73.845 %	69.823 %	75.735 %
L2R	71.315 %	67.297 %	73.204 %
ANALOG	71.077 %	66.625 %	73.169 %
R2L3	61.648 %	58.276 %	63.233 %
L2R3	53.276 %	49.672 %	64.912 %
zz	75.931 %	74.177 %	76.755 %
col	80.905 %	75.634 %	83.383 %
MST	83.984 %	82.230 %	84.809 %
CP	85.85 %	83.434 %	86.979 %

Table 1. The results of the Prague parsers (precision = recall)

	cross-brackets	precision	recall	LAA
all sentences				
Best trees	4.473	60.228 %	60.645 %	71.5 %
First trees	6.229	47.306 %	50.778 %	69.1 %
Average	5.799	45.627 %	46.584 %	69.0 %
projective sentences				
Best trees	3.619	66.718 %	68.663 %	73.1 %
First trees	5.289	53.028 %	57.630 %	70.6 %
Average	4.942	50.859 %	52.552 %	70.5 %
non-projective sentences				
Best trees	7.251	39.615 %	35.727 %	65.6 %
First trees	9.325	29.275 %	29.699 %	63.5 %
Average	8.625	29.112 %	28.097 %	63.3 %

Table 2. The results of the Brno parser on the e-test set

	cross-brackets	precision	recall	LAA
Best trees	0.792	89.519 %	92.274 %	97.2 %
First trees	2.132	70.849 %	74.358 %	92.6 %
Average	2.311	63.330 %	64.453 %	91.4 %
R2L		81.472 %		
L2R		81.634 %		
ANALOG		76.537 %		
R2L3		63.754 %		
L2R3		57.201 %		
zz		86.650 %		
col		90.129 %		
MST		89.889 %		
CP		91.912 %		

Table 3. The results of the Brno and Prague parsers on the small Brno tree set

In the case of phrasal trees we use the two following measures, PARSEVAL and leaf-ancestor assessment (LAA).

The PARSEVAL scheme utilizes only the bracketing information from the parser output to compute three values:

- *crossing bracket* – the number of brackets in the tested analyzer's parse that cross the tree bank parse.
- *recall* – a ratio of the number of correct brackets in the analyzer's parse to the total number of brackets in the tree bank parse.
- *precision*– a ratio of the number of correct brackets in the analyzer's parse to the total number of brackets in the parse.

There are several known limitations [16] of the PARSEVAL technique. It is not clear whether this metric can be used for comparing parsers with different degrees of structural fineness since the score on this metric is tightly related to the degree of the structural detail.

The leaf-ancestor assessment [17, 18] measure is more complicated than PARSEVAL. It considers a lineage for each word in the sentence, that is, the sequence of node-labels found on the path between leaf and root nodes in the respective trees. The lineages are compared by their edit distance, each of them having the score between 0 and 1. The score of the whole sentence is then defined as the mean similarity of the lineage-pairs for its respective leaves.

Since it considers not only boundaries between the phrases, the LAA measure is supposed to be more objective than the PARSEVAL, even at non-projective sentences. In this comparison we used the Geoffrey Sampson's LAA implementation, available at `http://www.grsampson.net/Resources.html`.

4.2 Problems and Discussion

Overall results of the Prague parsers testing are presented in the Table 1 in the form of percentage of correct dependendences for the whole set of sentences, for non-projecive and for projective only. The results of the Brno parser on the whole testing set (with manual tagging from PDT-2.0), e-test is displayed in the Table 2.

The experiment of comparing the results of parsers with dependency and phrasal outputs has opened several problems that we have tried to cope with. One of the main causes of these problems were the incompatibilities between the "phrasal PDT" trees and phrasal trees from the Brno parser. This was also the main source of low precision and recall of the parser. In order to prove this thesis, we have (manually) prepared a small set of phrasal trees[6] in the form of the Brno parser trees and repeated the measurements for this subset. The improvement of the results of the Brno parser on this small subset may be seen in the Table 3 and in the Figure 1.

5 Conclusions and Future Directions

In the paper, we have described a thorough comparison of the techniques and outputs of the two groups of parsers of the Czech language – the stochastic dependency Prague parsers and the meta-grammar phrasal Brno parser. We have summarized and discussed

[6] for 100 sentences randomly chosen from the e-test projective sentences

all the problems of a comparison of such different approaches and we have presented the measured results of the experiment. The results show that the Prague stochastic parser are better for general textual data, which do not have to follow (Czech) grammatical structures. However, it is not easy to give such conclusion for proper sentences.

In the future development, we would like to repeat this tests on another set of input data, namely on the prepared Brno phrasal tree bank. The question is whether this different testing set will shuffle the table of results significantly or it will stay more or less the same.

References

1. Hajič, J.: Building a syntactically annotated corpus: The Prague Dependency Treebank. In: Issues of Valency and Meaning, Prague, Karolinum (1998) 106–132
2. Horák, A., Kadlec, V.: New meta-grammar constructs in Czech language parser synt. In: Lecture Notes in Artificial Intelligence, Proceedings of Text, Speech and Dialogue 2005, Karlovy Vary, Czech Republic, Springer-Verlag (2005) 85–92
3. McDonald, R.: Discriminative learning and spanning tree algorithms for dependency parsing. PhD thesis, University of Pennsylvania (2006)
4. Hajič, J., Collins, M., Ramshaw, L., Tillmann, C.: A Statistical Parser for Czech. In: Proceedings ACL'99, Maryland, USA (1999)
5. Holan, T., Žabokrtský, Z.: Combining Czech Dependency Parsers. In: Lecture Notes in Artificial Intelligence, Proceedings of TSD 2006, Brno, Czech Republic, Springer Verlag (2006) 95–102
6. Holan, T.: Genetické učení závislostních analyzátorů. In: Sborník semináře ITAT 2005. UPJŠ, Košice (2005)
7. Holan, T.: Tvorba závislostního syntaktického analyzátoru. In: Sborník semináře MIS 2004. Matfyzpress, Prague, Czech Republic (2004)
8. Hall, K., Novák, V.: Corrective modeling for non-projective dependency parsing. (2005) 42–51
9. Nilsson, J., Nivre, J., Hall, J.: Graph transformations in data-driven dependency parsing,. In: Proceedings of the 21st Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney (2006) 257–264
10. Horák, A., Kadlec, V., Smrž, P.: Enhancing best analysis selection and parser comparison. In: Lecture Notes in Artificial Intelligence, Proceedings of TSD 2002, Brno, Czech Republic, Springer Verlag (2002) 461–467
11. Sedláček, R.: Morphemic Analyser for Czech. PhD thesis, Masaryk University (2005)
12. Hajič, J.: Disambiguation of Rich Inflection (Computational Morphology of Czech). Karolinum, Charles University Press, Prague, Czech Republic (2004)
13. Horák, A., Smrž, P.: Best analysis selection in inflectional languages. In: Proceedings of the 19th international conference on Computational linguistics, Taipei, Taiwan, Association for Computational Linguistics (2002) 363–368
14. Hajič, J.: Complex Corpus Annotation: The Prague Dependency Treebank, Bratislava, Slovakia, Jazykovedný ústav Ľ. Štúra, SAV (2004)
15. Collins, M.: dep2phr – conversion between dependency and phrase structures (1998) `http://ufal.mff.cuni.cz/pdt/Utilities/dep2phr/`.
16. Bangalore, S., Sarkar, A., Doran, C., Hockey, B.A.: Grammar & parser evaluation in the XTAG project (1998) `http://www.cs.sfu.ca/~anoop/papers/pdf/eval-final.pdf`.

17. Sampson, G.: A Proposal for Improving the Measurement of Parse Accuracy. International Journal of Corpus Linguistics **5**(01) (2000) 53–68
18. Sampson, G., Babarczy, A.: A test of the leaf-ancestor metric for parse accuracy. Natural Language Engineering **9**(04) (2003) 365–380