# Test Suite for the Czech Parser Synt

Vojtěch Kovář and Miloš Jakubíček

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
xkovar3@fi.muni.cz, xjakub@fi.muni.cz

**Abstract.** This paper presents a set of tools designed for testing the Czech syntax parser that is being developed at the Natural Language Processing Centre at the Masaryk University, synt. Testing the parser against a newly created phrasal tree corpora is very important for future development of the parser and its grammar. The usage of the test suite is not restricted to the synt parser but is open to wide scope of applications that provide similar output.

## 1 Introduction

Automatic syntactic analysis is one of the basic tasks in advanced natural language processing. However, the syntactic analysers (or parsers) developed for the Czech language deal with many serious problems, e.g. low precision or high ambiguity of the parsing results. For this reason, the development of the parsers must continue as effectively as possible and the qualities of the parsers must be continually tested against the corpus data.

This paper concerns a Czech parser synt that is being developed at the Natural Language Processing Centre at the Masaryk University (NLP Centre). The parser is based on context-free backbone with additional contextual actions and it features a developed meta-grammar formalism with a fast parsing algorithm. It produces sets of possible derivation phrasal trees and the output can be highly ambiguous. However, a tree-ranking algorithm is implemented that enables the parser to select one "best" tree from the output set in a short time that does not depend on the overall number of trees.

Until recently, there was no larger corpus of phrasal trees available. The only huge treebank for the Czech language was the Prague Dependency Treebank [1] but the dependency formalism is very different from the phrasal one and the conversion between dependency and phrasal structures can produce a large number of errors [2]. At the current time, a new treebank with phrasal trees has been built at the NLP Centre and we plan to use this treebank intensively in the process of the synt parser development.

In this paper, we introduce a set of tools (test suite) developed for testing the synt parser (as well as any other parser that produces similar outputs) using the new phrasal treebank. We briefly describe both the parser and the treebank and then we characterize the test suite itself: the procedure of testing, used metrics, comparison of a particular test with a reference one and related problems.

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008, pp. 63–70, 2008. © Masaryk University, Brno 2008

## 2 The `synt` Parser

The `synt` parser is based on a large Czech meta-grammar with context-free backbone and contextual actions. The involved parsing algorithm uses a modification of *head-driven chart parser* [3] that provides very fast parsing even in combination with big grammars. As mentioned in the introduction, the parser output produces set of ranked trees that match the parser meta-grammar.

Besides the parsing algorithm itself, many additional functions are implemented in the system, such as algorithm for finding the best coverage (for sentences that do not match the grammar), efficient selection of N best output trees from the analysis results or using so called *limits*.

The *limits* function is used if the user wants to prune the set of resulting trees according to their structure. The parser gets a set of limits on its input that can look like *0 4 np* and prints only the trees matching all the limits. In the previous example, only the trees would be printed in that a "np" (noun phrase) non-terminal covers the input from position 0 to position 4.

The coverage of the parser grammar is about 92 percent of Czech corpus sentences [4, p. 77]. Its precision was never rigorously evaluated because of insufficient syntactically annotated corpus data. (The only testing against a big corpus data is reported in [2] but the results indicate that the testing data were highly distorted by format conversions.) With the newly created phrasal treebank and test suite, we could make such evaluation. Its results are presented in the Section 4.6.

## 3 The Brno Phrasal Treebank

The Brno Phrasal Treebank was created in years 2006–2008 as a product of linguist specialists collaborating with the NLP Centre. The corpus contains in overall 86,058 tokens and 6,162 syntactically tagged sentences. The main source of sentences is the Prague Dependency Treebank.

Besides the correct tree in the phrasal formalism, the treebank source files contain information about the source of the text, lemmatized and morphologically tagged format of the text and limits that must be fulfilled by all correct trees. These limits contained in the treebank source files are used in one of the test suite statistics, as explained in following sections.

An example of a treebank sentence is shown in Figure 1.

## 4 The Test Suite

The test suite is a set of scripts that performs an automatic comparison of the `synt` parser output with the introduced phrasal treebank. Basically, it runs the parser over the morphologically tagged data from the treebank and incrementally computes the statistics according to the parser output.

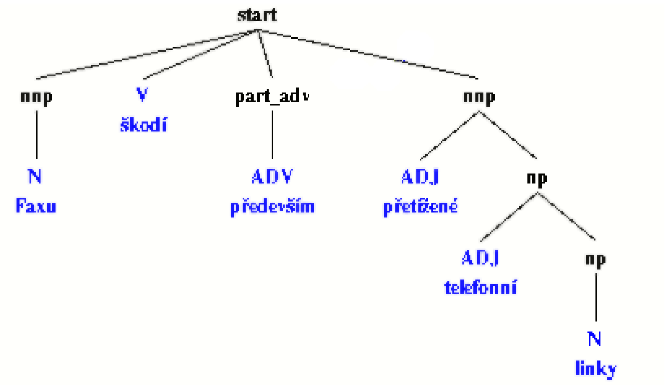


**Fig. 1.** An example of a treebank sentence

### 4.1 Included Statistics

The basic statistics we wanted to include in the testing are the following:

- *overall number of parsing trees* – useful for grammar ambiguity estimations.
- *number of limits trees* – or number of trees fulfilling limits. This number tells us how many "correct" trees have been found in the output. Ideally we want only one; if there are no such trees, the output of the parser is incorrect. In case of several trees, the limits recorded in the treebank should be probably more restrictive.
- *similarity* of the parsing results with the correct tree recorded in the treebank.

### 4.2 Measuring Similarity of Trees

The last of the presented statistics creates two questions:

- What similarity metric to use?
- How to handle ambiguous output of the parser with a tree-to-tree similarity metric?

Our answer to the first question is the usage of the metric called *leaf-ancestor assessment* (LAA) [5] proposed by Geoffrey Sampson in 2000. This metric is considered to be more reliable than the older PARSEVAL metric that is currently used more frequently. We outline the main characteristics of the metric in the following section.

The solution of the second problem is to use three different numbers for evaluation of the ambiguous output of the parser:

- *best tree similarity* – the best score of the LAA similarity metric reached by any tree from the output set.

- *average similarity* – average score of the LAA metric for all trees in the output set.
- *first tree similarity* – score of the best-ranked tree in the output set.

The first number can tell us how good the parser could be if we had an ideal tree-ranking function. The second one predicates of the overall precision of the grammar. The last item is probably the most useful since in most linguistic or NLP applications, we usually want *one best tree* from the parser, not a set; so this is the number that a potential user or advanced NLP application can expect when handling only one tree.

For efficiency reasons, we always take maximum 100 output trees as the whole output set.

Another complication related to the similarity measuring is the fact that the `synt` grammar, especially its set of non-terminals, slightly changes in time. For this reason, we applied *renaming* of the non-terminals in the resulting candidate trees as well as in the treebank trees. Moreover, the renaming of the non-terminals will make testing of other parsers by the same test suite possible and it can fix several small errors in the treebank data as well. The target set of nonterminals is shown in Table 1.

**Table 1.** Target non-terminals for renaming

| nonterminal | description |
|---|---|
| ABBR | abbreviation |
| ADJP | adjective phrase |
| ADVP | adverbial phrase |
| CLAUSE | clause |
| CP | conjunctions or punctuation (in the middle of sentence) |
| ENDS | ending sentence punctuation |
| NP | noun phrase |
| PP | prepositional phrase |
| PREP | preposition |
| PRON | pronoun |
| SENTENCE | the whole sentence (without ending punctuation) |
| VP | verb phrase |
| TOP | root nonterminal |
| OTHER | any other constituent (particle, interjection) |

### 4.3 The LAA Parse Evaluation Metric

Every possible parse evaluation metric has to compare two trees – the correct one (also called *gold standard*) and the one output by the parser (also called *candidate*). The LAA metric is based on comparing so called *lineages* of the two trees.

A *lineage* is basically a sequence of non-terminals found on the path from a root of the derivation tree to a particular leaf. For each leaf in the tree, the lineage is extracted from the candidate parse as well as from the gold standard parse. Then, the edit distance of each pair of lineages is measured and a score between 0 and 1 is obtained. The mean similarity of all lineages in the sentence forms the score for the whole analysis. More information about the metric can be found in [5].

In [6], it is argued that the LAA metric is much closer to human intuition about the parse correctness than other metrics, especially PARSEVAL. It is shown that the LAA metric lacks several significant limitations described also in [7], especially it does not penalize wrong bracketing so much and it is not so tightly related to the degree of the structural detail of the parsing results.

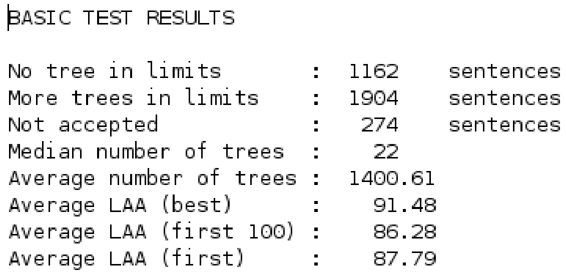
In the test suite, we used the implementation of the LAA metric by Derrick Higgins that is available at `http://www.grsampson.net/Resources.html`.

### 4.4 The output format

The results of each testing are saved in the form of a text file with 6 columns:

- sentence ID
- number of limits trees
- overall number of output derivation trees
- best tree similarity
- average similarity
- first tree similarity

After the test suite completes the whole file, a short summary is printed, as shown in the Figure 2.



```
BASIC TEST RESULTS

No tree in limits       :  1162    sentences
More trees in limits    :  1904    sentences
Not accepted            :   274    sentences
Median number of trees  :    22
Average number of trees :  1400.61
Average LAA (best)      :    91.48
Average LAA (first 100) :    86.28
Average LAA (first)     :    87.79
```

**Fig. 2.** The summary output of the test suite

### 4.5 Comparing Two Tests

During the parser development, we usually want to be able to compare several runs of the test suite in order to immediately gain a view of the impact of changes we have done. This enables us to prevent regressions in the development as well as it makes easier to track the changes history.

Thus, it is possible to perform a test-to-test comparison which outputs a table with test summaries. Furthermore, a detailed lookup of sentence changes is printed so that developers can directly correct any issues (see Figure 3). Currently, we collect following sentence differences (however the system is designed to be easily extended if further details were needed):

- sentences which do not pass the limits anymore
- sentences which newly cause a parser failure/timeout
- sentences with regressions in the number of trees/LAA values.

In order to speed up the comparison even more, an HTML document is produced as well, allowing the user (on-click) to obtain trees to compare after the tree images are created on-the-fly. A view of a tree-to-tree confrontation is provided in Figure 4.

### 4.6 Evaluation Results and Discussion

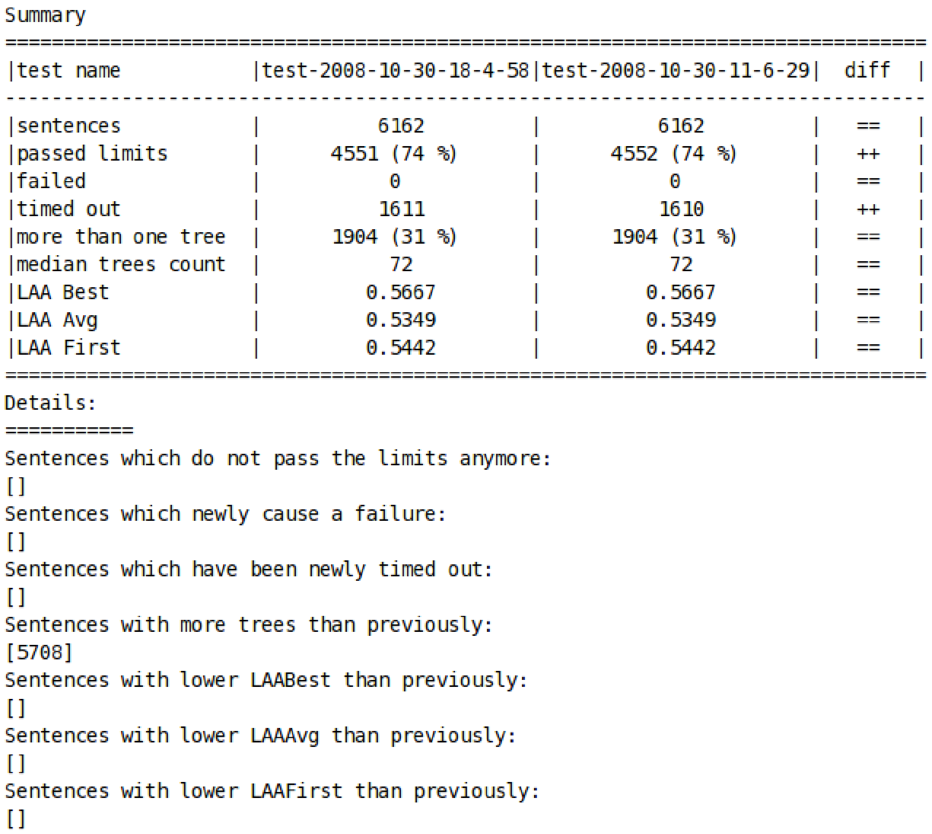
In the Figure 2, the results of a real test are shown. We can see that for 1,162 sentences (which is about 20 percent of the treebank) there is no correct tree in the parser output. However, the results of similarity measuring were relatively good – 87.8 percent for the first 100 trees. It can be also seen that the score for these first trees is better than average. This is a strong evidence that the parser ranking algorithm is basically correct. However, it could be still better; with an ideal ranking function we could reach the precision of 91.5 percent.

There is one remaining problem in interpretation of the results. For efficiency reasons, some parsing processes were killed during the testing since they exceeded a fixed time limit. It is an open question how to handle these "killed" sentences. In the evaluation presented above, these sentences were skipped and were not included into the statistic. If we counted them in with a score e.g. 0, the LAA metrics would fall down to 65–70 percent.

## 5 Conclusions and Future Directions

In the paper, we have presented a newly created test suite for the Czech parser `synt` that uses a new phrasal treebank for the Czech language. We have presented used metrics and procedures needed to get the results as well as outputs useful for developers of the parser. We also presented the precision of the parser measured by the introduced test suite.

In the future development, we mainly want to improve the parser grammar according to the data retrieved from the testing suite. At the same time, we plan to enhance the test suite according to the feedback we will get from the developers of the parser.



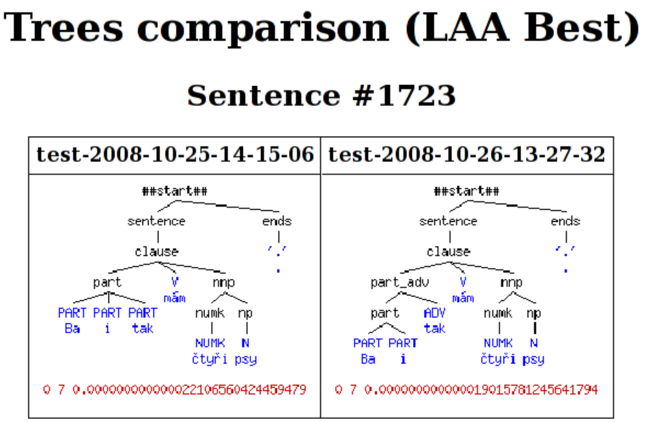
```
Summary
===============================================================================
|test name           |test-2008-10-30-18-4-58|test-2008-10-30-11-6-29|  diff  |
-------------------------------------------------------------------------------
|sentences           |         6162          |         6162          |   ==   |
|passed limits       |     4551 (74 %)       |     4552 (74 %)       |   ++   |
|failed              |          0            |          0            |   ==   |
|timed out           |         1611          |         1610          |   ++   |
|more than one tree  |     1904 (31 %)       |     1904 (31 %)       |   ==   |
|median trees count  |          72           |          72           |   ==   |
|LAA Best            |        0.5667         |        0.5667         |   ==   |
|LAA Avg             |        0.5349         |        0.5349         |   ==   |
|LAA First           |        0.5442         |        0.5442         |   ==   |
===============================================================================
Details:
===========
Sentences which do not pass the limits anymore:
[]
Sentences which newly cause a failure:
[]
Sentences which have been newly timed out:
[]
Sentences with more trees than previously:
[5708]
Sentences with lower LAABest than previously:
[]
Sentences with lower LAAAvg than previously:
[]
Sentences with lower LAAFirst than previously:
[]
```

**Fig. 3.** A test-to-test comparison output (random tests)

**Acknowledgements.** This work has been partly supported by the Academy of Sciences of Czech Republic under the projects 1ET100300414 and 1ET100300419 and by the Ministry of Education of CR in the National Research Programme II project 2C06009.

## 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., Holan, T., Kadlec, V., Kovář, V.: Dependency and Phrasal Parsers of the Czech Language: A Comparison. In: Proceedings of the 10th International Conference on Text, Speech and Dialogue, Pilsen, Czech Republic, Springer Verlag (2007) 76–84.
3. 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.



# Trees comparison (LAA Best)

## Sentence #1723

| test-2008-10-25-14-15-06 | test-2008-10-26-13-27-32 |
|---|---|
| 0 7 0.000000000000022106560424459479 | 0 7 0.000000000000019015781245641794 |

**Fig. 4.** A tree-to-tree confrontation selected in the HTML test-to-test comparison.

4. Kadlec, V.: Syntactic analysis of natural languages based on context-free grammar backbone. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno (2007).
5. Sampson, G.: A Proposal for Improving the Measurement of Parse Accuracy. International Journal of Corpus Linguistics **5**(01) (2000) 53–68.
6. Sampson, G., Babarczy, A.: A test of the leaf-ancestor metric for parse accuracy. Natural Language Engineering **9**(04) (2003) 365–380.
7. 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`.