***T. Jelínek***

# A SYSTEM FOR SYNTACTIC ANNOTATION OF LARGE CZECH CORPORA[1]

**Abstract.** We present a system of the pre-processing and post-processing of linguistic data leading to an improvement of stochastic dependency parsing results. We «condense» the data for the stochastic parser, i.e. we reduce the variability of word lemmas and forms in the text. After the parsing is done, we correct some of the recurrent parsing errors with a rule-based correction system. We achieve a 10.8% relative error reduction.

## Introduction

Syntactic annotation of a corpus is a useful resource allowing users to analyze linguistic phenomena that would be impossible to access using only morphological analysis. In our project, we deal with a dependency syntactic annotation for large Czech textual corpora. For such annotation, it is necessary to use fully automated tools. In this paper, we describe an improved system of stochastic dependency parsing. We use formalism and data from a manually annotated corpus, the Prague Dependency Treebank. With a linguistically designed pre- and post-processing of data, we achieve a significant improvement of the accuracy of stochastic parsing.

In the first part of this paper, we present the treebank and earlier experiments of stochastic parsing using its data. In the second part, we explain our choice of a parser and its original settings. The third part describes our system of pre-processing the data. The fourth part deals with a rule-based correction system. The final section summarizes the results and presents some ways to further improve the parsing of Czech.

[1] This paper resulted from the implementation of the project Czech National Corpus (LM2011023) funded by the Ministry of Education, Youth and Sports of the Czech Republic within the framework of Large Research, Development and Innovation Infrastructures.

## 1. Prague Dependency Treebank

The Prague Dependency Treebank[2] is a manually annotated corpus of written Czech with a multi-layer annotation. 1,5 million words (about 80,000 sentences) are annotated on the surface syntactic level (analytical layer)[3] with a dependency syntactic structure.

Fig. 1 shows the representation of the sentence: *Při posledních volbách se stal starostou. : In the last election he became mayor.* as a dependency tree in PDT.

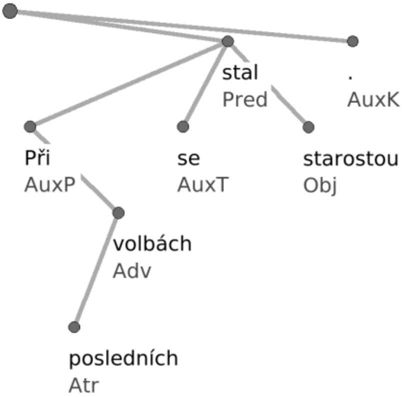

*Fig 1.* Example of a dependency tree in the PDT

PDT data are divided into the «training data» (80%) and two sets of test data (d-test and e-test, 10% each). Based on these data, many experiments of automatic parsing were undertaken. The best published result, with an accuracy of 85,8% obtained on the e-test, has been

[2] *Hajič J.* Complex Corpus Annotation: The Prague Dependency Treebank // Insight into the Slovak and Czech Corpus Linguistics. Bratislava, Slovakia, Veda, 2006. P. 54–73.

[3] See **http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/html/**

reached by Holan and Žabokrtský[4] with a combination of several parsers. The best results with a single parser have been achieved with the MaltParser[5] by D. Zeman[6] and with the MSTParser[7] by Novák and Žabokrtský[8]. Table 1 shows the unlabeled accuracy (UAS) of the parsers achieved on the e-test data.

*Table 1.* UAS achieved by parsers

| **Parser** | **Accuracy** |
|---|---|
| Combination | 85,8% |
| MaltParser | 85,6% |
| MSTParser | 84,7% |

## 2. MSTParser

For the parsing of large text corpora, we have chosen the MSTParser, although its published results are worse than those of the MaltParser, for it is about 25 times faster. An average speed of about 2,6s per sentence by the MaltPareser is hardly satisfactory for the annotation of large corpora. The MSTParser achieves an average speed of 0,11s per sentence.

[4] *Holan T., Žabokrtský Z.* Combining Czech Dependency Parsers // LNCS, 2006,Vol. 4188, TSD. Springer, Berlin/Heidelberg, Germany. P. 95–102.

[5] *Nivre J., Hall J., Nilsson J.* MaltParser: A Data-Driven Parser-Generator for Dependency Parsing // Proceedings of LREC2006, Genoa, Italy, 2006. P. 2216–2219.

[6] Unpublished, see **http://ufal.mff.cuni.cz/czech-parsing/**

[7] *McDonald R., Pereira F., Ribarov K., Hajič J.* Non-projective Dependency Parsing using Spanning Tree Algorithms // Proceedings of HLT/EMNLP, ACL, Vancouver, Canada, 2005. P. 523–530.

[8] *Novák V., Žabokrtský Z.* Feature Engineering in Maximum Spanning Tree Dependency Parser // LNCS, 2007, Vol. 4629, TSD Springer, Berlin/Heidelberg, Germany.

MSTParser in its original settings for Czech uses word forms and reduced morphological tags that contain only information about POS and case of the word, or POS and the subtype of POS, if case is not relevant (***P4*** for a pronoun in accusative, ***VB*** for a verb in the present form). The parser is unable to use effectively morphological tags containing more information (such as gender and number), they decrease its accuracy.

Thanks to better morphological disambiguation (used as input for parsing) performed by the morphological tagger Featurama[9], the accuracy of the parser was increased to 85,5% (e-test), achieving almost the same accuracy as the MaltParser.

| | | | | | | |
|---|---|---|---|---|---|---|
| *Při* | *posledních* | *volbách* | *se* | *stal* | *starostou* | *.* |
| R6 | A6 | N6 | P4 | Vp | N7 | Z: |
| 5 | 3 | 1 | 5 | 0 | 5 | 0 |
| AuxP | Atr | Adv | AuxT | Pred | Obj | AuxK |

*Fig. 2.* Format of data used by the MSTParser

### 3. The pre-processing of text

Stochastic parsers use training data to build a model of language that is later used to deal with new data. Parsing, however, faces the problem of sparse data: the low frequency of most words appearing in the training data does not allow the creation of reliable models that could use the forms or lemmas of these words. As the training data is necessarily limited, most of the words of a given language will never appear there at all. In the PDT data, for example, 30,000 different lemmas of nouns occur. However, in a large Czech text corpus SYN (1,3 bil. words, see http://www.korpus.cz), we find as much as 190,000 different lemmas of nouns.

[9] *M. Spousta*, see **http://sourceforge.net/projects/featurama/**

We solve a part of this problem by automatically pre-processing the data to «condense» them, e.g. to reduce data variability. We define narrow classes of words with identical syntactic properties using linguistic information from large corpora. In the data (both in training data and in the new texts to be parsed), we replace members of such classes with one representative of the class. Parser is then trained with such thickened data and new data are processed in the same way. Original forms and lemmas are restored after the parsing. Currently, some 80 word classes are defined. Some examples of these classes are listed below.

We use a list of several hundred names of towns divided by gender (*Praha*, *Varšava*, *Moskva ...*). In the text, we replace all the names of towns of the same gender by one representative (*Praha*).

We seek compound adjectives containing numbers (*dvacetiletý*, *osmapadesátiletý: 20/58-years-old*; *pětimetrový*, *stometrový: 5/100-meters-long*), we unify them replacing the number by one character (*xletý*, *xmetrový*).

Some semantic variants of indefinite pronouns (*jakýsi*, *bůhvíjaký*: *a kind of, God knows which*) are replaced with one basic indefinite pronoun (*nějaký* : *some*).

Ordinal numbers (*pátý*, *šestý*, *sedmý*...: *fifth*, *sixth*, *seventh*...) are replaced with one representative (*fifth*). We deal similarly with cardinal numbers, fractions, etc.

We create narrow classes of verbs based on their valency and aspect and replace them with one representative: for instance, *smát se*, *usmívat se*, *chechtat se*…: *laugh*, *smile*, *guffaw* (reflexive verbs without obligatory object valency, with potential dative or *na* + accusative valency) are replaced with one verb (*smát se*).

Overall, the number of various forms and lemmas in the training data is reduced by approximately 20%.

In order to further «condense» the data, we supply the parser with lemmas instead of word forms used in its original settings. Different forms of the same lemmas only rarely differ in syntactic properties and the variability of forms in the text is much higher than the variability of lemmas.

With this pre-processing of data, we achieved an increase in unlabeled accuracy of 0,6%, reaching 86,1% on the e-test data.

## 4. Rule-based correction

In a parallel research, independent of the pre-processing described above, we developed a rule-based system for the correction of errors in the output of stochastic parsing. A Czech text corpus (SYN2005) of 100 million words has been annotated syntactically using the original setting of MSTParser (accuracy 84,7%). In this corpus, we searched for frequently occurring types of errors of syntactic annotation. Whenever it was possible to design a reliable correction algorithm for a specific error type, we created a new correction rule.

The correction system reads the syntactically annotated text sentence by sentence; when it finds a typically erroneous structure, it invokes the appropriate correction rule. Some of the rules are heuristic; they choose the most likely correct structure. 28 such rules were created; new rules can be easily added.

In the annotated corpus, there is, for example, a frequent error of two uncoordinated subjects dependent on the same verb, as in the following examples. Errors in syntactic annotation are marked with (!).

1. *Bylo to*$_{\textbf{Sb}}$ *přiznání*$_{\textbf{Sb(!)}}$*porážky.*
   *It*$_{\textbf{Sb}}$ *was an admission*$_{\textbf{Sb(!)}}$ *of defeat.*

2. *Můj bratr*$_{\textbf{Sb}}$ *tam má obchod* $_{\textbf{Sb(!)}}$ *s diamanty.*
   *My brother*$_{\textbf{Sb}}$ *there has a shop*$_{\textbf{Sb(!)}}$ *with diamonds.*

3. *Tobě se to* $_{\textbf{Sb(!)}}$ *slovo*$_{\textbf{Sb}}$ *líbí?*
   *Do you like that* $_{\textbf{Sb(!)}}$ *word*$_{\textbf{Sb}}$*?*

In the examples 1 and 2, only the syntactic functions are wrong (***Sb*** instead of ***Pnom*** or ***Obj***). In the example 3, the word *to* is wrongly dependent on the verb *líbí* instead of the following noun *slovo*: the parser treated the word *to* as a syntactic noun, but here it should be treated as a syntactic adjective. In the example 2, the error of the parser is probably due to an error in morphological annotation (wrong case – accusative – assigned to the word *obchod*).

The rule targeting two uncoordinated subjects dependent on one verb has 10 variants of correction (algorithms), some of them correct only the syntactic label, others change the dependency structure, too. Their selection is influenced by several variables: the lemma of the verb (*být: to be* or another), gender and number agreement between the verb and both nouns labeled as subject, possible formal ambiguity (nominative – accusative) of both nouns, etc.

Overall, the correction system increased the unlabeled accuracy (UAS) by 0.4% from 84.7% to 85.1% and the labeled accuracy (LAS) by 1.3% from 77.6% to 78.9%.

When this correction system was tested on data annotated with the pre-processing described above, i.e. on data parsed with a significantly better accuracy, it proved much less effective. The increase in unlabeled accuracy was 0.1% (from 86.1% to 86.2%), below the statistical error; labeled accuracy increased by 0.4% (from 79.0% to 79.4%). With a better annotated input, the correction rules were applied less frequently and some of the stochastic rules even made more mistakes than performed due corrections.

## 5. Conclusion

We demonstrated two methods of increasing the accuracy of stochastic parsing: reducing variability in the data for parsing by an automatic pre-processing and a rule-based correction system. Using also a better morphological disambiguation, we were able to improve the unlabeled accuracy of the MSTParser from 84.7% to 86.2% (10.8% relative error reduction).

The most promising part of our combined system seems to be the «condensing» of data: it can be expanded by defining new classes of words with identical syntactic properties and implementing them in the system. A further careful reduction of variability both in training data and in new text to be parsed will increase both parsers accuracy and speed.

Our rule-based correction system must be adapted to better input data: existing correction rules will be refined, and new rules will be introduced. We do not expect, however, that this system could by itself contribute to a major increase in accuracy: the error variability in parsed text is too high and few correction algorithms are really reliable.