Adaptation of Czech Parsers for Slovak

Marek Medveď, Miloš Jakubíček, Vojtěch Kovář, Václav Němčík

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{xmedved1,jak,xkovar3,xnemcik}@fi.muni.cz

Abstract. In this paper we present an adaptation of two Czech syntactic analyzers Synt and SET for Slovak language. We describe the transformation of Slovak morphological tagset used by the Slovak development corpora skTenTen and r-mak-3.0 to its Czech equivalent expected by the parsers and modifications of both parsers that have been performed partially in the lexical analysis and mainly in the formal grammars used in both systems. Finally we provide an evaluation of parsing results on two datasets – a phrasal and dependency treebank of Slovak.

Key words: syntactic analysis, parsing, Slovak

1 Introduction

Czech and Slovak are both representatives of Slavonic free-word-order languages with rich morphology. The most differences between Czech and Slovak lie in the lexicon – on morphological and even more on syntactic level both languages are very similar. Currently, there is no full parser available for Slovak, only tools that produce partial analysis based either on regular expressions [1] or predicate-argument structure [2]. Because of the syntactic similarity of these languages, we took the opportunity to adjust two currently available Czech parsers, Synt and SET, for Slovak.

Syntactic analyzers Synt[3] and SET[4] have been developed over the past years in the Natural Language Processing Centre at Faculty of Informatics, Masaryk University. Both systems are rule-based but take a different approach to the challenges of syntactic analysis. The Synt parser is based on a context-free backbone enhanced with contextual actions and performs a stochastic agenda-based head-driven chart analysis. The syntactic parser SET is based on a simple grammar consisting of regular expressions over morphological tags and performs segmentation of input sentence according to the grammar rules.

The input for both Synt and SET is a sentence in the form of vertical text morphologically annotated by the morphological analyzer Ajka[5] which uses an attributive tagset described in [6][1].

[1] Current version of the tagset is available online at http:///nlp.fi.muni.cz/ma/.

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2012, pp. 23–30, 2012.

The output of Synt may be:

- **a phrase-structure tree**
  This is the main output of Synt and consists of a set of phrase-structure trees ordered according to the tree ranking. The system makes it possible to retrieve n-best trees effectively.
- **a dependency graph**
  A dependency graph represents a packed structure which can be utilized to extract all possible dependency trees. It is created by using the head and dependency markers that might be tied with each rule in the grammar.
- **set of syntactic structures**
  The input sentence is decomposed into a set of unambiguous syntactic structures chosen by the user.

The output of SET may be:

- **a hybrid tree** consisting of both dependency and constituent edges,
- **a pure dependency tree**,
- **a pure constituent tree**.

In our experiments, we used three Slovak corpora as input for the parsers – the r-mak 3.0[7] corpus, containing 1.2M tokens and manual morphological annotation and the skTenTen corpus[8], a large web corpus containing about 876M tokens with automatic morphological annotation, and a subset of a Slovak dependency treebank[9] that is currently under development in the Slovak Academy of Sciences, which contained more than 12,000 sentences and is further referred to as SDT.

For the parsers to be able to process the Slovak input, the following modifications had to be performed:

- **morphological tagging conversion** into the format expected by the parsers,
- **lexical analysis adjustment** in both parsers (e.g. mapping of lexical units to grammar non-terminals),
- **grammar adaptation** for both parsers, covering syntactic phenomena in which Czech and Slovak are different.

2 Morphological tagging conversion

In this section we describe the translation from the Slovak tagset to its counterpart in the Czech tagset and explain the steps necessary for correct function of syntactic analyzers. Both r-mak 3.0 and skTenTen use a positional tagset.[2] For the purpose of converting the annotation into the format given by the Czech morphological analyser Ajka, a translation script has been created called sk2cs.py, which takes a vertical text as input and translates each tag to its Czech equivalent.

[2] Available online at http://korpus.sk/morpho.html.

Obviously, there is no 1:1 mapping between tags in the tagsets, e.g. due to different subclassification paradigms for several part-of-speech (PoS) kinds. Therefore the translation process consists of three steps:

1. **rule-based translation**
2. **whitelist translation**
3. **whitelist completion**

2.1 Ruled-based tag translation

At first, the input tag is translated using a predefined set of rules that map each grammatical category to its counterpart in the Czech tagset. If this mapping is ambiguous (1:n), the program either just chooses the first tag or, optionally, produces an ambiguous output.

2.2 Whitelist-based tag translation

For words where the PoS of the Slovak tag is different than the one of its Czech equivalent, a whitelist-based procedure is used that directly maps selected words to their Czech tags. An example of a problematic translation is the word *mnoho* (a lot of) which is said to be an adverb in Czech but a numeral in Slovak. It should be noted that the morphological classification of this word (and many others) is a cumbersome issue with no clear solution, and we do not claim that the Czech or Slovak classification is better than the other one.

2.3 Whitelist-based tag completion

Finally, in some cases the Slovak tagset is less fine-grained than the Czech one and the resulting tag would not contain enough information for the parsing to be successful. This concerns e.g. pronouns for which the Slovak tagset does not contain any subclassification that would distinguish relative, interrogative and demonstrative pronouns, but both parsers use this kind of information in their grammar. Fortunately, the sets of all these pronouns are small enough to be handled case-by-case as well, and therefore the translation process uses another whitelist to extend the morphological annotation for them.

3 Adaptation of `Synt`

`Synt` is a rule-based parser consisting of a context-free grammar (CFG) enhanced by in-programmed contextual actions for capturing contextual phenomena like e.g. grammatical agreement. The parsing process consists of two steps: first a basic chart parsing is performed using the CFG and producing a large set of candidate analyses in the form of the resulting chart – a packed forest of trees. On top of the chart, the contextual actions are evaluated, pruning the analyses space by orders of magnitude and producing final parsing results.

To prevent maintenance issues a rule-based system may suffer from, the grammar is developed in the form of a meta-grammar, consisting of only about 250 rules. From this meta-grammar a full grammar is automatically derived by exploiting per-rule defined derivation actions (e.g. expanding a group of non-terminals or permutating right-hand side of a meta-rule).

The modifications for Slovak in `Synt` consist from two parts:

- **adaptation of lexical analysis**
- **adaptation of grammar rules**

3.1 Lexical analysis adaptation

In `Synt` lexical analysis is a process that assigns a pre-terminal (i.e. last non-terminal in the tree that is directly rewritten to the surface word) to a given word by using word's morphological classification. In some cases (e.g. identification of some named-entities like months, or specific handling of modal verbs), the lexical analysis exploits not only the tag of the word, but also its lemma or the word itself. In these cases the analysis had to be modified (translated) to Slovak.

3.2 Grammar rules adaptation

In the following we list a number of syntactic phenomena that need to be handled differently in Czech and Slovak.

Sentences with passive Expression of passive in Slovak is different from Czech. The Czech passive structure is: ***to be + passive verb*** (figure 1). But in Slovak the structure is: ***to be + adjective***.

```
clause %> is vpasr
vpasr -> VPAS
```

Fig. 1. Original rule for passive.

Therefore it is necessary to adapt this rule (figure 2). The adaptation consists of replacing pre-terminal `VPAS` by pre-terminal `ADJ`.

```
clause %> is vpasr
vpasr -> ADJ
```

Fig. 2. Adapted rule for passive in Slovak language.

Sentences with structure *not + to be* This structure shows the main difference between Slovak and Czech. In Slovak (figure not2) this structure is expressed by two words but in Czech language it is expressed only by one word.

```
Original:          clause %> IS sth
                   clause %> ARE sth
                   clause %> VB12 sth
                   --------------------
Adapted:           clause %> is sth
                   clause %> are sth
                   clause %> vb12 sth
                   is -> IS
                   is -> IS NOT
                   are -> ARE
                   are -> ARE NOT
                   vb12 -> VB12
                   vb12 -> NOT VB12
```

Fig. 3. Adaptation of Czech rule for Slovak structure ***not + to be***

Sentences with structure *would + to be* The same case as structure ***not + to be*** is the structure ***would + to be***. The modification of this rule (figure4) divides one word into two words with same semantics.

```
Original:          clause %> VBK sth
                   -----------------
Adapted:           clause %> vbk sth
                   vbk -> VBK
                   vbk -> VBK VB12
```

Fig. 4. Structure ***would + to be***

Sentences with structure *if + to be* or *that + to be*
The next case of Slovak structure which contains two divided words instead of one word expression is structure ***if + to be*** or ***that + to be***. The new rule describing this two structures is on figure 5.

Sentences with multiple numbers For sentences with structure „three times“ there was no pre-terminal for word „times“ which is written separately in Slovak. A new rule associated with this pre-terminal was created too. This new rule can analyzed structure ***„three times“***, or structure ***„3 times“***(figure 6).

```
Original:          clause %> akvbk sth
                   akvbk -> KVBK
                   akvbk -> AVBK
                   -------------------
Adapted:           clause %> akvbk sth
                   akvbk -> KVBK is
                   akvbk -> AVBK is
                   akvbk -> KVBK are
                   akvbk -> AVBK are
                   akvbk -> KVBK vb12
                   akvbk -> AVBK vb12
```

Fig. 5. Rule for structure ***if + to be*** and ***that + to be*** (sk)

```
numk -> NUMK TIMES
```

Fig. 6. Added pre-terminal TIMES

3.3 Adaptation of SET

SET is based on a simple grammar mostly consisting of regular expressions over morphological tags. Similarly to Synt, the grammar is directly lexicalized in some cases and required appropriate modifications. Besides the lexical analysis, following changes have been performed to the grammar:

Structure *would + to be*
In the same way as in Synt, the Czech expression for this structure had to be divided into two words (figure 7).

```
TMPL: $PARTICIP $...* $BY $BYBYT MARK 2 DEP 0    PROB 1000
%$BYBYT(word): som si sme ste
%TMPL: $BY $BYBYT MARK 1 DEP 0 PROB 1000
```

Fig. 7. Structure ***would + to be***

Structure *not + to be*
The same situation as before is in this case (figure 8).

4 Evaluation

The modifications have been evaluated for both parsers separately. For Synt, the coverage was measured on two corpora, the r-mak 3.0 and SDT. To convert

```
%TMPL: $PARTICIP $...* $NOT $BYBYT MARK 2 DEP 0    PROB 1000
%$BYBYT(word): som si sme ste
```

Fig. 8. Structure ***not + to be***

the SDT treebank from its native XML format into annotated vertical text, the `pdt2vert`[10] was used. The precision of `Synt` was measured on a random sample of 77 sentences from the `skTenTen` corpus that were accepted by the parser and for which a correct constituent tree was determined. The LAA tree similarity metric [11] was used for the evaluation.

Since `SET` always produces some dependency tree, only dependency precision was evaluated against the SDT.

4.1 Evaluation of `Synt` parser

Corpus	Number of sentences	Number of accepted
r-mak 3.0	74,127	77 %
SDT	12,762	76.9 %

Table 1. Evaluation of the coverage of `Synt`

Number of sentences	77
Median number of trees	148
Average number of trees	71595.81
Average LAA of the first tree	87.13
Time per sentence	0.038 s

Table 2. Evaluation of the precision of `Synt`

4.2 Evaluation of `SET` parser

Corpus	Number of sentences	Dependency precision
SDT	12,762	56.7 %

Table 3. Evaluation of the precision of `SET`

5 Conclusions and Future Development

In this paper we have presented two Czech parsers, `Synt` and `SET`, adapted for Slovak. These represent first full parsing solutions available for Slovak. In the future further development of both parsers on Slovak is planned towards better precision and coverage on larger datasets.

Acknowledgements

We hereby thank Radovan Garabík, Ľudovít Štúr Institute of Linguistics, Slovak Academy of Sciences for his kind help and willingness to provide us with development and evaluation data. The work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Trabalka Marek, B.M.: Realization of syntactic parser for inflectional language using XML and regular expressions. In: Text, Speech and Dialogue, volume 1902 of Lecture Notes in Computer Science, (Springer Berlin / Heidelberg) (2000) pages 59–90
2. Ondáš Stanislav, Juhár Jozef, and Čižmár Anton: Extracting sentence elements for the natural language understanding based on Slovak national corpus. In: Analysis of Verbal and Nonverbal Communication and Enactment. The Processing Issues, volume 6800 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg (2011) 171–177
3. Jakubíček, M., Horák, A., Kovář, V.: Mining phrases from syntactic analysis. In: Text, Speech and Dialogue. (2009) 124–130
4. Kovář, V., Horák, A., Jakubíček, M.: Syntactic analysis using finite patterns: A new parsing system for czech. In: Human Language Technology. Challenges for Computer Science and Linguistics, Berlin/Heidelberg (2011) 161–171
5. Šmerk, P.: Fast Morphological Analysis of Czech. In: Proceedings of the Raslan Workshop 2009, Brno (2009)
6. Jakubíček, M., Kovář, V., Šmerk, P.: Czech Morphological Tagset Revisited. Proceedings of Recent Advances in Slavonic Natural Language Processing 2011 (2011) 29–42
7. Ľudovít Štúr Institute of Linguistics, Slovak Academy of Sciences: Slovak National Corpus – r-mak3.0. (2009) `http://korpus.juls.savba.sk/`, [Online].
8. Masaryk University, Lexical Computing Ltd.: skTenTen – Slovak web corpus (2011) `http://trac.sketchengine.co.uk/wiki/Corpora/skTenTen`, [Online].
9. Gajdošová, K.: Syntaktická anotácia vybraných textov slovenského národného korpusu. In Múcsková, G., ed.: Varia. 16. Zborník materiálov zo XVI. kolokvia mladých jazykovedcov, Slovak Linguistic Society, Ľudovít Štúr Institute of Linguistics, Slovak Academy of Sciences (2009) s. 140 – 148
10. Němčík, V.: Extracting Phrases from PDT 2.0. In Horák, A., Rychlý, P., eds.: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2011, Brno, Tribun EU (2011) 51–57
11. Sampson, G., Babarczy, A.: A test of the leaf-ancestor metric for parse accuracy. Natural Language Engineering **9**(04) (2003) 365–380