Partial Grammar Checking for Czech Using the SET Parser

Vojtěch Kovář

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

Abstract. Checking people's writing for correctness is one of the prominent language technology applications. In the Czech language, punctuation errors and mistakes in subject-predicate agreement belong to the most severe and most frequent errors people make, as there are complex and non-intuitive rules for both of these phenomena. At the same time, they include numerous syntactic, semantic and pragmatic aspects which makes them very difficult to be formalized for automatic checking. In this paper, we present an automatic method for fixing errors in commas and subject-predicate agreement, using pattern-matching rule-based syntactic analysis provided by the SET parsing system. We explain the method and present first evaluation of the overall accuracy.

Keywords: parser, SET, Czech, grammar checking, punctuation detection, syntactic analysis

1 Introduction

Reliable checking people's writing for correctness is one of the important goals in natural language processing. Spelling checkers became a common part of our lives, but checking more complex language phenomena still presents a challenge. Although there are "grammar checkers" available in software packages like Microsoft Office or as stand-alone programs, they can address only a restricted range of grammar error types, and they are far from being able to find all the errors, wisely following the "minimum number of false alerts" philosophy.

In the Czech language, punctuation errors and mistakes in subject-predicate agreement belong to the most severe and most frequent errors people make, as there are complex and non-intuitive rules for both writing punctuation and correct usage of subject-predicate agreement.[1] At the same time, these rules include numerous syntactic, semantic and pragmatic aspects which makes them very difficult to be formalized for automatic checking.

Punctuation detection and fixing errors in the Czech grammar is often used as a textbook example of how automatic syntactic analysis can be exploited for a prominent

[1] In Czech, subject-predicate agreement is difficult mainly because of homophonic verb endings (i/y) and differences between standard and colloquial language. E.g. "psi štěkali" ("dogs barked") is correct, "psi štěkaly" is wrong (but it reads the same), "děvčata šla" ("girls went") is correct, but very frequent colloquial form "děvčata šly" is wrong.

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 308–314, 2014.

practical application. However, in real life, the full parsing is rarely used (and if, the results are not convincing [1]), and the current methods use rather various types of common error patterns or light-weight modifications of the full syntactic formalisms.

In this paper, we introduce case studies of new methods for punctuation correction, and detection of subject-predicate agreement violations in Czech. Both of the studies exploit syntactic parser SET [2].

2 Related Work

There are two commercial systems for grammar checking of Czech: The Grammar checker built into the Microsoft Office, developed by the Institute of the Czech language [3], and the Grammaticon checker created by the Lingea company [4]. Not much has been published about the principles these are based on; most of the available materials are Czech-only and have rather advertising character. According to available information, both tools are trying to describe negative (wrong) constructions and minimize number of false alerts, i.e. prefer precision over recall significantly (frequent false alerts bother users and make them stop using the tool). The available tests of these tools [5,6] (available only in Czech) indicate that the tools are able to fix 25-35 percent of errors, with the number of false alerts around 6-30 percent.

The Czech parsing community also contributed to the grammar checking problem. Holan et al. [1] proposed using automatic dependency parsing, however, authors conclude that the results have only a prototype character and much work is still needed to achieve practically usable product. Jakubíček and Horák [7] reported on using the Synt parser [8], together with a specialized grammar for Czech to detect punctuation in sentences. They report over 80 percent precision and recall in punctuation *detection* which means that the system fills in the commas into the text without commas (rather than into a text with errors). 80 percent in detection roughly means that every fifth comma is missing and every fifth is wrong. It is not completely clear how the system would behave on real erroneous texts and it is not possible to re-test, as the tool is not available at the moment.

3 The SET Parser

The SET parsing system,[2] firstly introduced in [2], was designed according to the principles of agile and rapid software development [9,10] that we adopt in our solutions, too. Namely, design simplicity and practical usability was the highest priority that was taken into account in all phases of development, rather than accuracy compared to the data annotated according to linguistic theories.

The core of the SET system is formed by a pattern matching engine and a variant of maximum spanning tree algorithm. The tool is open source and its distribution contains several sets of pattern matching rules ("grammars"), the default one being the grammar for parsing general Czech. The rule syntax is illustrated in Figure 1, and explained more

[2] SET is an abbreviation of "syntactic engineering tool"

```
TMPL: verb  …  $AND  …  verb   MARK 0 2 4 <coord>  HEAD 2
    $AND(word): , a ani nebo
```

Fig. 1. Example of a SET rule, describing coordination of two verbs using one of the Czech conjunctions *a*, *ani*, *nebo* (and, neither, or), or a comma, with any gap between the verbs and the conjunction. If the rule is matched and selected, the relevant tokens are to be marked as a coordination in the tree, with the conjunction being the head of this constituent.

in detail in [2], or on the SET project page.[3] The primary output of the system, *hybrid tree*, combines dependency and constituent structure features (in form of special phrasal tokens inserted into a dependency tree), and allows conversion into pure dependency or pure constituent structure formalisms.

4 Punctuation Detection

We have designed a specialized SET grammar for punctuation detection, together with an added special output function which prints a comma before each word marked by a special phrasal token (we used <c>, as illustrated in the examples). The grammar contains 10 rules for analysis of the most important patterns where a missing punctuation should be added, that are used for building a reduced tree where the only important information are the tokens marked with <c>. The rules are dealing with following phenomena:

- commas between coordination members (2 rules)
- relative clause boundaries (6 rules)
- 1 particular type of apposition (1 rule)
- 1 rule is negative and specifies where the comma should not be written before relative pronoun (which is normally a clause boundary)

This approach is deliberately approximative, and follows the more straightforward pattern matching idea of Grammaticon and Grammar checker, rather than the full syntactic analysis introduced by Jakubíček and Horák [7]. However, it is one of our future goals to combine the added functionality with the full power of the standard SET grammar and compare the results with the shallow approach.

Examples of a punctuation rule, a reduced syntactic tree for a sentence with missing punctuation, and the resulting sentence with completed punctuation, are given in Figures 2 and 3. As we can see, the "syntactic tree" on the SET output contains practically no syntactic information, except the <c> guidelines for completing the sentence punctuation – rather than that, the SET parser is used as an economical pattern matching engine.

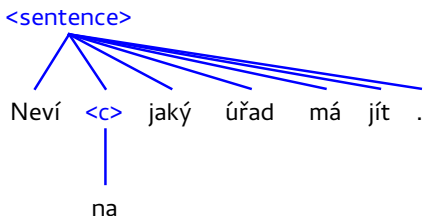
Evaluation of the functionality was performed using the Desam corpus [12], using the same methodology as Jakubíček and Horák [7] – deleting all commas from the input sentences and comparing the original texts with the output of the parser.

[3] nlp.fi.muni.cz/projects/set

```
TMPL: $NEG $PREP $REL  MARK 1 <c>  HEAD 1
    $REL(tag): k3.*y[RQ] k6.*y[RQ]
    $PREP(tag): k7
    $NEG(tag not): k7 k3.*y[RQ] k6.*y[RQ] k8
    $NEG(word not): a * " tak přitom
```

Fig. 2. One of the punctuation detection rules in SET, matching preposition (`k7`) and relative pronoun (`k3.*y[RQ]`) or adverb (`k6.*y[RQ]`), not preceded by preposition or conjunction (`k8`) or relative pronoun/adverb and few other selected words (the `tag not` and `word not` lines express negative condition – token must not match any of the listed items). Ajka morphological tagset is used [11].

```
Input: Neví na jaký úřad má jít.
```




```
Output: Neví, na jaký úřad má jít.
```

Fig. 3. Illustration of SET punctuation analysis – reduced tree and the output sentence with completed punctuation. The rule from Figure 2 was matched. Sentence: *"Neví na jaký úřad má jít."* (missing comma before *"na"* – *"(He) does not know what bureau to go in."*).

The results are summarized in Table 1. We have distinguished a sample of first 500 sentences from the corpus, and the whole corpus of 50,000 sentences; also, we worked with both automatic and correct manual morphological tagging. We can see that the results are very similar for all the testing sets, and it can be concluded that errors in automatic tagging do not influence punctuation detection significantly.

The system shows very high, nearly 95 percent precision, which is very good as it minimizes the number of false alerts. Recall is rather low which means that the system is able to find only about 50 percent of errors. Speed of the analysis was in all cases rather high – 313 sentences per second, on a single Intel Xeon 2.66 GHz core.

We have performed a manual investigation of the differences between the parser output and the correct punctuation, on first 150 sentences of the testing data. This insight showed that many of the parser errors are actually not errors – in Czech, in some places the comma is not necessary but writing it is not a mistake. Out of the missed commas, 21.4 percent were not necessary according to the Czech writing rules (most frequent real errors were in coordinations). From the false positives, 50 percent were actually correctly placed commas. If we extrapolate these percentages to the whole Desam testing set, we get the numbers as in the Extrapolation row.

Table 1. Results of punctuation detection within the SET system.

Testing set	Precision (%)	Recall (%)	F-measure (%)
Desam 500 (manual tagging)	94.7	47.3	63.1
Desam full (manual tagging)	94.1	45.0	60.9
Desam 500 (automatic tagging)	95.3	45.4	61.5
Extrapolation	97.1	56.8	71.6

5 Subject-Predicate Agreement

Unlike the previous case study, detecting errors in subject-predicate agreement in Czech sentences uses the standard full SET grammar. The rules detecting subjects of clauses (labelling them as "subject" and adding their dependency on the verb) were differentiated to correct subjects that agree with the detected verb in gender and number, and the salient candidates for subject that do not fulfill the agreement condition. The latter ones were labelled as "subject-bad", for marking the difference. Example of the respective SET rules is given in Figure 4, and the output trees are illustrated in Figure 5.

```
TMPL: $MAINVERB $...* $LIKESUBJ   AGREE 0 2 gn
        MARK 2  DEP 0  PROB 602  LABEL subject
TMPL: $MAINVERB $...* $LIKESUBJ
        MARK 2  DEP 0  PROB 601  LABEL subject-bad

   $MAINVERB(tag): k5.*mF ...
   $LIKESUBJ(tag) $LIKESUBJ(tag): k1.*c1 k3.*c1.*xP ...
```

Fig. 4. One of the SET subject rules, and its twin detecting bad subject-predicate agreement. The main difference is the AGREE action associated with the first rule, which enforces agreement in gender (g) and number (n). `MAINVERB` and `LIKESUBJ` are common variable definitions for both rules.

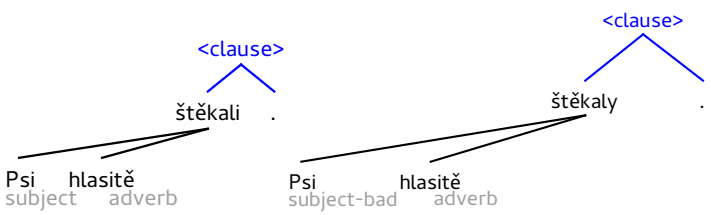


Fig. 5. SET output tree for correct and incorrect version of sentence *"Psi hlasitě štěkali." ("Dogs loudly barked.").*

Again, the current rules within the SET grammar cover the most frequent patterns. There are more complicated cases where the subject consists of a complex coordination,

error in which would not be detected by our solution, in certain cases. However, according to the YAGNI principle[4] which is part of rapid application development philosophy, we first implement and test the straightforward approach, then identify the real drawbacks and then plan how to fix them, rather than devising a complete solution at the beginning and suppose that we are able to anticipate possible problems. Correctness of the YAGNI principle showed very early in this case.

As there is no large available database of frequent Czech subject-predicate agreement errors, we have decided to use a small set of sentences from a Czech primary school dictation, where frequent errors were manually identified and classified [13]. The set contained 26 sentences with 11 subject-predicate errors. Although the testing set is small, from Table 2 we can clearly see that there is a problem in automatic morphological tagging. The difference in recall between the manual and automatic version is immense, and the reason is that the subjects in the erroneous clauses were tagged as non-subjects, e.g. as accusative instead of nominative (there is very frequent nominative-accusative homonymy in Czech), and therefore they were not recognized as subjects by the parser. This is probably caused by the fact that the tagger (Desamb [14]), as it is usual for taggers, was trained on correct texts and the non-agreement between subject and predicate is so rare in these texts, that it chooses rather another option. Actually, most of the tagging errors resulted in syntactically correct Czech sentences, sometimes even semantically correct, although not suitable in the given context. This is a complex problem that will require a new approach to Czech tagging.

Table 2. Results of subject-predicate agreement checking within the SET system.

# sentences	26
# errors	11
# errors spotted (automatic tagging)	2 (18%)
# false alerts	0
# errors spotted after tagging correction	7 (64%)

Another problem are sentences with unvoiced subject (usually present in the previous sentence) – this was in 3 of the 11 sentences. Solution to this problem requires quality anaphora resolution, and we did not attempt to solve it within this case study.

Notable is the 100 percent precision that we have obtained in case of both manual and automatic tagging – there was no false alert.

6 Conclusions

Our system for **punctuation detection**, using as few as 10 rules, outperforms the general reported results for Grammaticon and Czech grammar checker, in terms of both precision and recall – number of false alerts below 3% is very good compared to them, and the recall is better as well. Jakubíček and Horák [7] reported better recall but lower precision; and we are confident that the precision is more important here, due to the

[4] `en.wikipedia.org/wiki/You_aren't_gonna_need_it`

bothering character of false alerts, and any tool with precision lower than 90–95 percent is not suitable for practical usage. Thanks to its results, our tool is ready to be built into a grammar checking application.

The **subject-predicate agreement** case study revealed a serious problem in automatic tagging of erroneous Czech texts. Nevertheless, thanks to the 100 percent precision, the system can be immediately employed in a grammar checker as well.

Because of the specific Czech writing rules, the proposed grammar modifications cannot be used for other languages without further changes. However, the approach proved very expressive – it is able to produce good results with very few rules, so it should be straightforward to adapt it for other languages.

Acknowledgement

This work has been partly supported by the Ministry of Education of the Czech Republic within the LINDAT-Clarin project LM2010013.

References

1. Holan, T., Kuboň, V., Plátek, M.: A prototype of a grammar checker for Czech. In: Proceedings of the 5th conference on Applied natural language processing, Association for Computational Linguistics (1997) 147–154
2. 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. Lecture Notes in Computer Science, Berlin, Springer (2011) 161–171
3. Oliva, K., Petkevič, V., Microsoft s.r.o.: Czech grammar checker (2005) `office.microsoft.com/word`.
4. Lingea s.r.o.: Grammaticon (2003) `www.lingea.cz/grammaticon.htm`.
5. Pala, K.: Pište dopisy konečně bez chyb – Český gramatický korektor pro Microsoft Office. Computer (2005) 13–14
6. Behún, D.: Kontrola české gramatiky pro MS Office - konec korektorů v Čechách? (2005) `interval.cz/clanky/kontrola-ceske-gramatiky-pro-ms-office-konec-korektoru-v-cechach`.
7. Jakubíček, M., Horák, A.: Punctuation detection with full syntactic parsing. Research in Computing Science, Special issue: Natural Language Processing and its Applications **46** (2010) 335–343
8. Horák, A.: Computer Processing of Czech Syntax and Semantics. Librix.eu, Brno, Czech Republic (2008)
9. Martin, J.: Rapid application development. Macmillan (1991)
10. Gabriel, R.P.: Lisp: Good news, bad news, how to win big. AI Expert **6** (1991) 30–39
11. Sedláček, R., Smrž, P.: A new Czech morphological analyser ajka. In: Proceedings of Text, Speech and Dialogue, 4th International Conference. Lecture Notes in Computer Science, Berlin, Springer (2001) 100–107
12. Pala, K., Rychlý, P., Smrž, P.: DESAM — annotated corpus for Czech. In: Proceedings of SOFSEM'97, Berlin, Springer (1997) 523–530
13. Trifanová, B.: Analýza chyb v diktátech žáků po absolvování 1. stupně ZŠ. Bachelor thesis, Masaryk University (2014) `is.muni.cz/th/382965/ff_b`.
14. Šmerk, P.: Unsupervised learning of rules for morphological disambiguation. In: Proceedings of Text, Speech and Dialogue, 7th International Conference. Lecture Notes in Computer Science, Berlin, Springer (2004) 211–216