Human Language Technologies – The Baltic Perspective
I. Skadiņa and R. Rozis (Eds.)

doi:10.3233/978-1-61499-701-6-147

A New Phase in the Development of a Grammar Checker for Latvian

Daiga DEKSNE[1]
Tilde, Latvia

Abstract. The paper reports on the recent work in the development of a grammar checker for Latvian. The grammar checker is using extended context free grammar (CFG) formalism for description of correct and erroneous syntactic structures. The grammar checking engine uses both of these sets of the rules. The grammar checker is used by language learners as well as native speakers. Our recent work is directed at the creation of an error-annotated corpus of texts that are created by non-native speakers. Based on this corpus, the CFG rule set is refined.

Keywords. Error classification, grammar checking, Latvian language, language learners

1. Introduction

For the past two decades, from time to time, our company has been returning to the further development of the proofing tools for Latvian. We started with spell checking of a single word. Grammar checking is a much more complicated task, especially for highly inflected languages such as Latvian. Our first generation grammar checker was developed using an advanced pattern matching technique. A certain sequence of tokens in a sentence was searched, and conditions on values of morphological features were checked. This technique did not allow the description of long distance agreement errors and errors describing complex syntactic structures, it was quite slow, and many rules matched false errors because of high morphological ambiguity.

For our second generation grammar checker, we extended CFG (context free grammar) formalism by adding syntactic roles, lexical constraints, and constraints on morpho-syntactic feature values [1], [2]. The formalism allowed the attribution of morpho-syntactic feature values to phrases and specification of optional components. The grammar checker was implemented by using two sets of rules – rules describing correct sentences and rules describing grammatical errors. With this technique, even very complex syntactic structures could be parsed.

An indispensable part of the development of a grammar checker is the analysis of errors made by users. We have created an error annotated corpora for grammar checker assessment. As a result of the empirical analysis of texts, we have defined 22 types of errors that can be grouped into five larger groups – formatting errors, orthography errors, morphology and syntax errors, punctuation errors, and style errors. The collected corpora has two parts – the student paper corpus (which is split into two parts:

[1] Corresponding Author: Daiga Deksne, SIA Tilde, Vienibas gatve 75A, Riga, Latvia, LV-1004; E-mail: daiga.deksne@tilde.lv

one part for the development of rules and the other part for quality assessment purposes) and the balanced corpus. In total, there are 20,877 sentences [3]. During the previous stage, we achieved 0.702 precision and 0.275 recall on the student paper development corpus and 0.650 precision and 0.187 recall on the student paper test corpus.

Our more recent efforts have been directed at addressing the needs of specific user groups and at analyzing the errors that the grammar checking system fails to recognize in already collected corpora. In this paper, we give a short overview of the architecture of the grammar checker for Latvian and the extended CFG rule formalism. We describe the creation and annotation of the non-native speaker corpus and compare errors found in it to errors in a previously created corpus.

2. System Architecture and Rule Formalism

Approaches for the grammar checking task differ. The most popular are rule based systems that use formal grammars for rule description. For Swedish [4] and Norwegian [5] grammar checkers, constraint grammar formalism is used; it was originally designed by Fred Karlsson [6]. There are various formalisms derived from the classical context free grammar formalism, which was formalized by Chomsky [7] and Backus [8]. Two examples are Generalized Phrase Structure Grammar [9] and Definite Clause Grammar [10]. Other formalisms are also based on phrase structure, for example, Head-Driven Phase Structure Grammar [11], Lexical Functional Grammar [12], etc. Along with the rule based approaches, statistical [13] and hybrid approaches [14] coexist. The grammar checking task can be viewed as a statistical machine translation (SMT) task that translates erroneous text into the correct text. Such SMT systems are reported in CoNLL-2014 Shared Task [15], where the grammar checkers correct English essays written by second language learners of English. In recent years, neural machine models are gaining popularity in solving the grammatical error correction task [16], [17].

We use a rule based approach. As an inflected language, Latvian requires a large number of non-terminals for representation of morpho-syntactic features. For example, in a noun phrase, an adjective and a noun must have the same gender, number, and case. In Latvian, there are masculine and feminine cases, singular and plural numbers, and seven case values. Using the original CFG formalism, 28 rules are required for description of a simple noun phrase consisting of an adjective and a noun (see Figure 1).

```
NP -> A_masc_sg_nom N_masc_sg_nom
NP -> A_masc_pl_nom N_masc_pl_nom
NP -> A_fem_sg_nom N_fem_sg_nom
NP -> A_fem_pl_nom N_fem_pl_nom
NP -> A_masc_sg_gen N_masc_sg_gen
NP -> A_masc_pl_gen N_masc_pl_gen
NP -> A_fem_sg_gen N_fem_sg_gen
NP -> A_fem_pl_gen N_fem_pl_gen
…
```

Figure 1. An example of rules describing a noun phrase in original CFG formalism.

To avoid this problem, we have extended the CFG formalism by defining constraint and assignment operators to be used in a rule body (Figure 2 shows an example of the rule). A rule must have a description line similar to classical CFG: the production rule has a single non-terminal symbol on the left side and one or several terminal and/or non-terminal symbols on the right side. The rule has a rule body in which constraint operators allow to check the morpho-syntactic properties of terminals and non-terminals. With the help of assignment operators, the left-side non-terminal of the rule can inherit some properties from the right-side constituents.

```
NP -> attr:A main:N
  attr:A.Case==main:N.Case
  attr:A.Number==main:N.Number
  attr:A.Gender==main:N.Gender
  NP.Case=main:N.Case
  NP.Number=main:N.Number
  NP.Gender=main:N.Gender
```

Figure 2. An example of a rule describing a noun phrase in extended CFG formalism.

Rules describing an error are a little bit different (see Figure 3 for an example of the rule describing error in agreement). They have a markup operator stating error boundaries, an error description line, and instructions for generation of suggestions. They use assignment operators to change the incorrect values of rule constituents. A name of the left-side non-terminal is in the form of 'ERROR-id'; it cannot inherit properties from the right-side constituents.

```
DESCR "Disagreement error"
ERROR-1 -> attr:AP main:NP
    Disagree(attr:AP,main:NP, Case, Number, Gender)
GRAMMCHECK MarkAll
    attr:AP.Case=main:NP.Case
    attr:AP.Number=main:NP.Number
    attr:AP.Gender=main:NP.Gender
SUGGEST(attr:AP+main:NP)
```

Figure 3. Example of an error rule describing disagreement in case, number, or gender.

For parsing, the Cocke-Younger-Kasami (CYK) algorithm [18] is used. Since ungrammatical sentences cannot be fully parsed, this algorithm also allows partial parsing. The extended CFG formalism of the rules are described in detail in [2].

3. Analyzing Errors of Specific User Groups

Through empirical studies of annotations of bachelor or master theses written in Latvian by students from different disciplines (computer science, Russian philology, and history), we have noticed a distinction in the error types common for every group and in the overall quality of the texts. 36.55% of sentences written by computer science students, 31.43% of sentences written by history students, and 60.29% of sentences written by students of Russian philology contained errors. Punctuation and writing

style errors are common to all students. Many students of Russian philology are non-native Latvian speakers. This group would benefit from a grammar checker that is able to detect and correct specific spelling and syntactical errors. Such errors are common to language learners whose native language does not have the long vowel, the palatalized consonant sounds, or some morphological categories of words, for example, the definiteness of the adjectives. In the *Learner Corpus Bibliography*[2] that is maintained by the Centre for English Corpus Linguistics at the Catholic University of Louvain, there are numerous references to the error-annotated language learner corpora for English and some other languages. Unfortunately, there are no large-scale learner corpora for Latvian. There are some publications analyzing various types of errors in contemporary Latvian, particularly in the language of media, in scientific papers of students, and in texts found on web sites. A concise handbook of Latvian grammar [19] contains examples of errors typical to foreign students taking Latvian language classes. A small corpora contains texts that have been written by Baltic language learners of Lithuanian or Latvian background [20].

4. A New Corpus of Error Annotated Data

We have created a corpus of written text with errors made by non-native Latvian speakers. The sentences are collected from bachelor or master theses of Russian philology students and from various sources on the web. The corpus contains 679 sentences in total. We have annotated the sentences using a predefined set of error types. We have refined some existing types of errors and defined some new types of errors that have not been used in previous error annotated corpora. The new types are the following: wrong choice of pronoun, inappropriate use of the reflexive verb, wrong choice of number for noun (singular instead of plural and vice versa), wrong choice of prefix, wrong choice of tense for verb. We make the distinction between misspelled words (regardless of their context) and words that are not appropriate for the context. In language, there are words that might differ in spelling by one or two letters. By mistyping a letter, a different correct word that is contextually inappropriate could be written. In such a case, the usage of an incorrect word might only be detected by analyzing the surrounding context. The distribution of the various types of errors for the new non-native speaker corpus is shown in Table 1. The data for the student paper development corpus, which was created during a previous stage of the grammar checker development, is included for comparison.

Table 1. Various types of errors found in different corpora (%).

Error type	Student paper development corpus	Non-native speaker corpus
Wrong choice of prefix	-	2.80
Formatting errors	7.38	-
Word not appropriate for context or loan translation	5.31	17.97
Misspelled word	4.85	16.20
Words to be written together	1.47	1.62
Wrong choice of pronoun	-	0.74
Inappropriate use of reflexive verb	-	0.44
Capitalization error	4.55	0.74
Comma error in a subordinate clause	15.36	5.01
Comma error in a participial clause	8.44	2.36

[2] at http://www.uclouvain.be/en-cecl-lcbiblio.html

Error type	Student paper development corpus	Non-native speaker corpus
Comma error in an insertion	2.58	1.33
Comma error in a grouping	0.71	-
Unmotivated comma usage	5.15	10.60
Wrong type of sentence	0.35	-
Division of equal parts of a sentence	3.84	0.15
Missing dash error	2.37	-
Wrong sequence of sentence parts	0.51	3.83
Error in word sequence	3.28	-
Wrong location of a preposition	0.05	-
Definite/indefinite ending usage for adjectives	3.94	11.49
Wrong mood of verb	1.26	1.62
Error in negation	0.51	-
Wrong case of a noun	0.81	4.42
Preposition requires different case of a noun	-	0.44
Wrong choice of number for noun (singular/plural)	-	1.18
Wrong choice of tense for verb	-	0.44
Agreement between several subjects and a predicate	-	0.29
Agreement error in nominal phrases	12.99	3,39
Style error	10.51	12.96
Unspecified error	3.79	-

5. Results and Discussion

The most popular individual types of errors found in compared corpora are different. In the student development corpus, the error types exceeding 10 percent of the total error count in the corpus are comma errors in subordinate clauses, agreement errors, and style errors. In the non-native speaker corpus, five error types exceed 10 percent of the total error count – word not appropriate for context or loan translation, misspelled word, unmotivated comma usage, definite/indefinite ending usage for adjectives, and style errors. These error types reveal the details of the language that are the hardest for non-native speakers of Latvian to learn.

Table 2. Evaluation results.

Corpus	Recall		Precision		F-measure	
	previous	recent	previous	recent	previous	recent
Non-native speaker	-	0.468	-	0.935	-	0.624
Student paper (dev.)	0,275	0.352	0.702	0.732	0.396	0.475
Student paper (test)	0.187	0.273	0.65	0.768	0.291	0.402

We have refined the rule set used by our grammar checker to detect the new error types in the non-native speaker corpus. This new rule set is also used for the previously created corpora. In order to evaluate the quality of the grammar checker in general and for certain error types specifically, we calculate recall, precision, and f-measure [21]. There are good results for the new non-native speaker corpus, and there is notable improvement for the existing corpora (see Table 2).

At the moment, the phrases with a contextually inappropriate word or loan translations are included in a dictionary. This is not a very effective way to deal with such errors. Some statistical module should be trained using a big text corpora for assessing how well a word fits a given context.

References

[1] D. Deksne, R. Skadiņš, CFG Based Grammar Checker for Latvian. In *Proceedings of the 18th Nordic Conference of Computational Linguistics NODALIDA 2011*. (2011), 275–278.

[2] D. Deksne, I. Skadiņa, R. Skadiņš, Extended CFG formalism for grammar checker and parser development. In *Computational Linguistics and Intelligent Text Processing*. Springer Berlin Heidelberg, (2014), 237–249.

[3] D. Deksne, I. Skadina, Error-Annotated Corpus of Latvian. In *Baltic HLT*. (2014), 163–166.

[4] A. Arppe, Developing a grammar checker for Swedish. In *12th Nordic Conference in Computational Linguistics (Nodalida-99)*, Trondheim, (2000), 13–27.

[5] K. Hagen, J. B. Johannessen, P. Lane, Some problems related to the development of a grammar checker. In *NODALIDA '01, the 2001 Nordic Conference in Computational Linguistics*. (2001), 21–22.

[6] F. Karlsson, Constraint Grammar as a framework for parsing running text. In *13th International Conference on Computational Linguistics (COLING-90)*, vol. 3, Helsinki (1990), 168–173.

[7] N. Chomsky, *Syntactic structures*. Mouton, The Hague (1957).

[8] J.W. Backus, The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference. In: *International Conference on Information Processing*, UNESCO (1959), 125–132.

[9] G. Gazdar, *Generalized Phrase Structure Grammar*. Harvard University Press (1985).

[10] F. Pereira, D. Warren, Definite clause grammars for language analysis--A survey of the formalism and a comparison with augmented transition networks. In *Artificial Intelligence*, vol. 13(3), (1980), 231–278.

[11] C. Pollard, I. A. Sag, *Head-driven phrase structure grammar*. University of Chicago Press, Chicago (1994).

[12] R. M. Kaplan, J. Bresnan, Lexical-Functional Grammar: A formal system for grammatical representation. In: Bresnan, J. (ed.) *The Mental Representation of Grammatical Relations*, Cambridge, MA, The MIT Press (1982), 173–281.

[13] J. Sjöbergh, O. Knutsson, Faking errors to avoid making errors: Very weakly supervised learning for error detection in writing. In: *Recent Advances in Natural Language Processing IV* (RANLP 2005), Borovets (2004), 506–512.

[14] J. Xing, L. Wang, D. F. Wong, S. Chao, X. Zeng, UM-Checker: A Hybrid System for English Grammatical Error Correction. In: *17th Conference on Computational Natural Language Learning* (CoNLL-2013), vol. 34 (2013).

[15] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, C. Bryant, The CoNLL-2014 Shared Task on Grammatical Error Correction. In *CoNLL Shared Task* (2014), 1–14.

[16] Z. Yuan, T. Briscoe, Grammatical error correction using neural machine translation. In *Proceedings of NAACL-HLT*. 2016, 380–386.

[17] C. Sun, X. Jin, L. Lin, Y. Zhao, X. Wang, Convolutional Neural Networks for Correcting English Article Errors. In *National CCF Conference on Natural Language Processing and Chinese Computing*. Springer International Publishing, 2015, 102–110.

[18] D. H. Younger, Recognition and parsing of context-free languages in time n 3. *Information and control*, 10(2), 1967, 189–208.

[19] A. Rubīna, *Latviešu valodas rokasgrāmata: valodas kultūra teorijā un praksē*. Rīga : Zvaigzne ABC, 2005.

[20] I. Znotiņa, Learner corpus Esam: a new corpus for researching Baltic interlanguage. In: *Corpus Linguistics 2015*. Abstract book. Lancaster: UCREL, 2015, 447-448.

[21] C. J. Van. Rijsbergen, Evaluation. In: *Information Retrieval* (2nd ed.). Newton, MA, USA: Butterworth, 1979.