Developing a constraint grammar for Lithuanian

Trey Jagiella
Indiana University Bloomington
fjagiell@indiana.edu

1 Introduction

This paper presents a preliminary constraint grammar for Lithuanian. The main objective in developing this constraint grammar was precision. The corpus used to develop this constraint grammar with the Lithuanian ALKSNIS treebank from the Universal Dependencies Project (Bielinskiene et al., 2016). The pipeline consists of a morphological analyser of all possible interpretations for the wordforms in the corpus as well as a constraint grammar. In the test corpus, the constraint grammar has a precision of .9205, a recall of .1845, and an F1 score of .3074.

The paper is organized as follows: section 2 contains a brief review of literature, section 3 describes the analysis pipeline, section 4 describes the development process, section 5 evaluates the results, and section 6 presents the conclusion.

2 Review of literature

There has not yet been a constraint grammar developed for the Lithuanian language. There has, however, been a fair amount of linguistic research on the language. Since the fall of the Soviet Union, there has been greater study into other areas of the Lithuanian language. Little of this work, however, has been translated into other languages (Usoniene et al., 2012). Nevertheless, there are English-language books and translations on Lithuanian grammar as well as Lithuanian dictionaries readily available, which were consulted in the development of this constraint grammar (Mathiassen, 1997; Ramoniene and Pribusauskaite, 2008; Piesarkas, 2006; Piesarkas and Svecevicius, 1995).

3 Analysis pipeline

3.1 Morphological analyser

To create a list of wordforms and their possible interpretations, I used a Python program, which creates a list of all interpretations for a single wordform found in an input.

3.2 Rule writing

The constraint grammar, lt.cg3, is composed of 79 rules, 26 of which are remove and 53 of which are select. These rules were developed by running a sentence of the train CoNLL-U file through the morphological analyser to find a list of its outputs. Based on the ambiguities of the output and the correct lemma in the corpus, rules were written to make the constraint grammar pick the correct interpretation.

4 Development process

To test the rules in the constraint grammar and further develop it a script was run. It took the dev CoNLL-U file of the Lithuanian ALKSNIS treebank and ran it through the analyser and lt.cg3 files and then compared the output of this process to the annotation in the dev file. This script outputs the true and false positives for each rule; the number of input, output, and reference analyses; input, output, and reference ambiguity; total true and false positives and negatives; and precision, recall, and F-score.

From the rule by rule output of the script, poorly performing rules could be eliminated or modified accordingly.

5 Evaluation

5.1 Corpus analysis

The next table summarizes the ambiguity left in the test corpus after being run through the constraint grammar:

Ambiguity in the test corpus

	Input	Reference	Output
Analyses	12629	10118	10932
Ambiguity	1.25	1.0	1.08

From the table above, it can be seen that while there was a noticeable reduction in ambiguity, whether correctly or incorrectly, there still remains a large portion ambiguity within the corpora.

The following table demonstrates the performance of the constraint grammar through precision and recall.

Precision and Recall

	dev	test
Precision	.87	.92
Recall	.66	.18
F1 Score	.75	.31

As can be seen in the table above, the rules are performing relatively accurately. However, the recall is still fairly low, particularly in the test corpus.The F1 score shows that overall, the constraint grammar has a modest performance in disambiguating the corpus.

In the following table the number of true and false positives and negatives for the test corpus are presented:

Positives and Negatives in the test corpus

	Positives	Negatives
True	1563	3367
False	135	6915

From the table above, we can see that the number of true positives is significantly higher than the number of false positives. On the other hand, this is not the case for true and false negatives.

The difference in recall between corpora is intriguing. Given the fact each corpus is relatively small, just over 10,000 tokens each, there is plenty of room for variability and building rules using one corpus may not cover the language sufficiently to allow for effective rule making.

6 Conclusion

Although the precision is not bad at 92% in the test corpus, with a recall of 18% , there is still much work to be done. Any future rules written should be more accurate than the current ones in addition to the rewriting of current rules to increase their accuracy.

References

Agne Bielinskiene, Loic Boizou, Jolanta Kovalevskaite, and Erika Rimkute. 2016. Lithuanian Dependency Treebank ALKSNIS. *I. Skadia and R. Rozis (Eds.): Human Language Technologies The Baltic Perspective*.

Terje Mathiassen. 1997. *Short Grammar of Lithuanian*. Slavica Publishers, Inc., Columbus, OH.

Bronius Piesarkas. 2006. *Didysis Lietuviu-Anglu Kalb Zodynas*. Zodynas Publishers, Vilnius, Lithuania.

Bronius Piesarkas and Bronius Svecevicius. 1995. *Lithuanian Dictionary English-Lithuanian Lithuanian-English*. Zodynas Publishers, Vilnius, Lithuania.

Meilute Ramoniene and Joana Pribusauskaite. 2008. *Practical Grammar of Lithuanian*. Baltos Lankos, Lithuania.

Aurelija Usoniene, Nicole Nau, and Ineta Dabasinskiene. 2012. *Multiple Perspectives in Linguistic Research on Baltic Languages*. Cambridge Scholars Publishing, Newcastle upon Tyne, UK.