Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/978-1-61499-442-8-111

Dependency Parsing of Estonian: Statistical and Rule-based Approaches

Kadri MUISCHNEK [a,1], Kaili MÜÜRISEP [a] and Tiina PUOLAKAINEN [a]
[a] *University of Tartu, Estonia*

Abstract. This paper gives an overview of the latest developments in computational syntactic analysis of Estonian. We present Estonian Dependency Treebank, an ongoing corpus annotation project. Although the treebank construction is still under way, we have used it for training MaltParser and experimenting with combining MaltParser with a rule-based Constraint Grammar parser for Estonian. MaltParser achieves unlabeled attachment score (UAS; correct links to head node) of 83.4% and label accuracy (LA) of 88.6%. Labeled attachment score (LAS) was 80.3%.

Applying different algorithms for combining MaltParser with Constraint Grammar parser improved the results by 1%. Special CG rule set for fixing some typical MaltParser errors improved the UAS by up to 1.5%.

Keywords. Estonian syntax, dependency parsing, treebank

Introduction

The syntactic analyzer of Estonian [1] is based on the Constraint Grammar (CG) formalism [2] and its latest version uses VISL CG-3 format and software [3]. The analyzer consists of separate sets of hand-crafted grammar rules for determining clause boundaries, morphological disambiguation, syntactic mapping and syntactic function assignment. A set of rules for recognizing dependency relations is currently under development. Recent syntactic tagging of a large corpus of written Estonian showed that the recall of shallow syntactic analysis is 92.6% and precision 72% [4]. This means that more than one quarter of words in the text remain syntactically ambiguous.

In recent years there have been efforts for creating Estonian dependency treebank. In June 2014 it consisted of nearly 400,000 manually annotated words, covering the text classes of fiction, newspaper and scientific texts.

During the last decade, the interest for developing dependency parsers has increased considerably. The dependency parsing seems to be the best method for syntactic analysis of morphologically rich languages with relatively free word order like Estonian.

We use the open-source system MaltParser [5] for our parsing experiments with the Estonian Treebank. To facilitate MaltParser optimization we use MaltOptimizer [6] - a system that automates the search for optimal parameters based on the analysis of the training set. The evaluation was conducted using MaltEval tool [7].

[1]Corresponding Author: Institute of Computer Science, University of Tartu, Liivi 2, Tartu, 50409, Estonia; E-mail: kadri.muischnek@ut.ee.

When comparing the results from both parsers, we found that the combination of parsers will increase the performance even more.

The rest of the paper is organized as follows. Section 1 provides an overview of related work and Section 2 describes the annotation scheme of our treebank. Sections 3 and 4 report the experimental results of applying MaltParser to Estonian and describe the experiments of combining rule-based and probabilistic parsers. In section 5, we conclude the paper with discussing some ideas for future work.

1. Dependency Treebanks

This section gives a limited overview of similar projects. It is impossible to give an exhaustive overview in a short article, so we have considered only work on languages closely related to Estonian or being relevant for the Baltic HLT audience.

Finnish, a language closely related to Estonian, has two larger-scale treebank projects, both representing the dependency structure.

FinnTreeBank is a manually annotated dependency treebank based on 17,000 model sentences in the Large Grammar of Finnish [8]. The FinnTreeBank annotation scheme [9] is shallow, the dependency structure is based on the word-forms actually occurring in the text, no virtual nodes (e.g. for ellipsis) are postulated. What makes the annotation scheme somewhat different is its semantic orientation, meaning that the semantically laden words are analysed as the governors and words expressing mainly grammatical relations as dependents, e.g. an adposition is governed by the noun, not vice versa. The semantic principle causes some important divergences from the latest academic grammatical description of Finnish [8], e.g. FinnTreeBank annotation scheme recognizes adessive subjects in possessive and cognizer clauses.

The annotation scheme of Turku Dependency Treebank [10] is a Finnish-specific version of the well-known Stanford dependency scheme [11]. The annotation consists of two layers; the first layer is based on the standard version of Stanford annotation scheme, with language-specific modifications; the annotation in the second layer covers phenomena like propagation of conjunct dependencies, external subjects, syntactic functions of relativizers. Differently from the original Stanford scheme and the FinnTreeBank annotation scheme, a virtual node is inserted for annotating gapping, a special type of ellipsis.

Latvian Treebank [12] has been annotated using a hybrid model in relation to dependency and phrase structure grammars. A special concept of x-word has been introduced, that can act as a governor or as a dependant, but is essentially a phrase structure node. This virtual node is used as a governor of multi-word named entities, verbal chains, coordinations and adpositional phrases. In case of ellipsis, the omitted element is "restored" using a virtual node. While converting the hybrid treebank into a pure dependency treebank, special experiments were conducted for deciding, which annotation to choose; for example whether for verb chains the auxiliary, modal or lexical verb should be annotated as the governor.

The annotation scheme of the Lithuanian Treebank [13] distinguishes five basic grammatical relations, namely those of subject, object, predicative, attribute and modifier plus an additional underspecified dependency relation. Lithuanian, like Estonian has a rich morphological system, the POS tagset contains 18 different categories and the tagset for morphological features includes 12 different categories, whereas the number

of morphological tags assigned to one word-form varies from 0 to 9. So, there is actually no need for more detailed set of syntactic labels as the information is present in the combination of morphological and syntactic labels.

Joakim Nivre and Ryan McDonald [14] have constructed an Universal dependency annotation scheme for annotating any human language with dependency structures. Their starting point has again been the Stanford annotation scheme [11]. Using available treebanks in several languages that used slightly modified versions of the Stanford annotation scheme (as for example the aforementioned Turku Dependency Treebank), they merged language-specific labels used only in one-two languages into more general ones resulting in more general annotation scheme hopefully suitable for annotating typologically diverse languages. Still, there are two types of constructions where the annotation may vary across the languages: adposition phrases and copula constructions.

One of the main differences between the proposed Universal dependency scheme and ours' is, again, that we do not annotate clausal functions like clausal subject, we simply chain the clauses by attaching a governing verb of the complement clause to the governing verb of the main clause, the only exception being the relative clauses that are attached to the noun they are modifying.

2. The Dependency Treebank of Estonian and its Annotations Scheme

The Dependency Treebank of Estonian is an ongoing annotation project, the aim of which is to create a 400,000-word syntactically annotated corpus by the end of the year 2014 and to have all the texts double-annotated, parallel annotations compared and discrepancies solved by a super-annotator. By June 2014 we had almost annotated the planned amount of texts and were concentrating on comparing the annotated versions and solving the discrepancies.

The treebank is annotated using the Estonian Constraint Grammar tagset. The annotation has separate layers for morphology, surface syntax and dependency relations.

The morphological annotation layer contains information about lemma, part of speech and grammatical categories (e.g. case and number for nominals; mood, tense, person and number for verbs) for every word-form in text[2]. Also, the valency information has been added to the records of some word-forms.

Surface-syntactic layer contains the syntactic function labels. According to our annotation scheme, the members of the verbal chain can have labels FMV (finite main verb), IMV (infinite main verb), FCV (finite chain verb), ICV (infinite chain verb). Particles as parts of a particle verb are tagged Vpart, and if the particle verb is a nominalization, then the particle has a tag VpartN. The verb negator is labelled as NEG.

The arguments of the verb are labelled as subject SUBJ, object OBJ, predicative PRD or adverbial ADVL; the adjuncts also get the adverbial ADVL label.

The attributes of a nominal are tagged according to their word-class: AN stands for adjectival attribute, NN for nominal attribute and apposition, DN for adverb attribute, INFN for infinitival attribute and KN for an adpositional phrase as an attribute (label is attached to the adposition as it is considered to be the governor of the adpositional phrase, the noun governed by an adposition receives a label P). A word-form governed

[2] A table containing all the morphological tags can be found here:
http://www.cl.ut.ee/korpused/morfliides/seletus.php?lang=en

by a quantor is labelled as Q. The premodifying and postmodifying labels have been distinguished by adding arrow symbols to them (AN> is premodifying adjectival attribute, <NN is postmodifying nominal attribute). J stands for conjunctions and I for interjections. The main shortcoming of our annotation scheme is that we do not distinguish between adverbial modifiers and adverbial complements.

The syntactic layer is shallow, meaning that no virtual nodes are postulated. Dependency layer gives information about the governor of every word-form in text.

In general, our annotation scheme is quite coarse for annotating intra-clausal phenomena, and comparable e.g. to the Stanford annotation scheme. It should also be kept in mind that a lot of information that the Stanford tagset presents explicitly in the form of syntactic labels, we present as a combination of morphological and syntactic labels. For example, we do not distinguish between coordinating and subordinating conjunctions on the level of syntactic labels, but this information is present at the morphological level, where there are two different POS-labels: J crd and J sub.

However, while annotating the dependency relations that hold between the clauses we do not distinguish clausal subjects, clausal complements or clausal modifiers, we only state that there is a dependency relation between the clauses.

An example in Figure 1 demonstrates the use of tags in Estonian CG format. Word-forms are in separate rows followed by their morphological and syntactic description. The description consists of the lemma, ending, POS, morphological information, valency information (in angle brackets), syntactic label (starting with @) and dependency information (starting with #). The first word-form *Kliendina* is a substantive (S), common noun (com), plural (pl), in essive case (es), starting with capital letter (cap), it is functioning as an adverbial and depends on the word-form in the position 2 (#1->2).

```
"<Kliendina>"                                                  % client
    "klient" +na S com sg es cap @ADVL #1->2
"<huvitab>"                                                    % interests
    "huvita" +b V main indic pres ps3 sg ps af <Part> @FMV #2->0
"<mind>"                                                       % me
    "mina" +d P pers ps1 sg part @OBJ #3->2
"<mugavus>"                                                    % convenience
    "mugavus" +0 S com sg nom @SUBJ #4->2
"<ja>"                                                         % and
    "ja" +0 J crd CLBC @J #5->7
"<asjaajamise>"                                                % procedure
    "asja_ajamine" +0 S com sg gen @NN> #6->7
"<selgus>"                                                     % clarity
    "selgus" +0 S com sg nom @SUBJ #7->4
"<.>"
    "." Z Fst #8->7
```

Figure 1. Sample sentence "As a client, I am interested in the convenience and clarity of the procedure."

3. Statistical Parsing

Due to the rapid growth of Estonian treebank, we have started to consider the use of statistical parsers, and have selected MaltParser for our treebank. We have chosen MaltParser [5] since it has been successfully employed for a wide range of languages, including morphologically rich languages with relatively small treebanks (for example, Latvian

and Lithuanian). In addition, MaltParser includes the MaltOptimizer system [6] which helps the end user to select appropriate parameters and parsing algorithm without having expert knowledge on underlying methods.

Estonian Treebank uses Constraint Grammar format, i.e. textual format with almost unlimited tagset. As MaltParser allows to use fine-grained POS tags besides the usual ones, some morphological information has been merged to POS tags. For example, proper names have their own label, we use prepositions and postpositions instead of adpositions etc. As the regular set of POS tags consists of 15 tags, there is also an opportunity to employ 22 fine-grained POS tags in CoNNL format. Most of morphological description has been retained except valency information (a large number of rare tag combinations). The syntactic annotation remain same as in the EstCGParser annotation (27 labels), except that the main verb of the main clause (or the head of the verbless clause) gets the label ROOT. We do not annotate the functions of whole clauses. The trees in EstCG format were converted to CoNNL format. Only the double-checked part of the whole treebank has been used for statistical parsing which consisted of 191,000 tokens, 13,310 sentences. Half of the corpus consists of newspaper texts (95,000 tokens), while the other half contains fiction (46,000 tokens) and scientific texts (49,000 tokens). All the sentences have been manually morphologically disambiguated. Every 5th sentence was moved to the testing part of corpora (37,959 tokens), so the training set consists of 153,471 tokens.

First, we used MaltOptimizer to find most appropriate training model and parameters. The tool suggested to use Covington-Non-Projective algorithm and a specific feature model.

The entire learning process of MaltParser took 3 minutes and 47 seconds while the parsing of the test corpus was completed in 7 seconds (we used a Linux desktop with Intel Core i7-4770 CPU and 16 GB of memory).

The preliminary results gave labeled attachment score (LAS, the label and relation link are both correct) 83.6% on 37,959 tokens. This result includes the analysis of punctuation marks (which is a trivial task) and non-sentential constructions like passages in foreign languages, chemical formulas or bibliographical references in scientific texts annotated by label @NONE. We excluded punctuation marks and non-sentential constructions from the analysis, the LAS decreased to 80.3% (31,434 tokens). Also, we observed the unlabeled attachment score (UAS) of 83.4% and the label accuracy (LA) of 88.6%.

When we compare these experiments with results from our previous research, the shallow EstCGParser of written text analyzes 88 - 90% of words unambiguously and its error rate is 2% (if the input is morphologically disambiguated and error-free). These results have been achieved mainly on the corpora of fiction texts, and are similar to LA metric (also, the analysis of punctuation marks has been excluded). The handmade grammar has never been adopted for scientific texts nor newspaper texts with a lot of financial information. When we applied the EstCGParser for the current corpora and evaluated it with the same MaltEval tool, its label accuracy was merely 85.8%. If the parser would be configured to leave some of the analyses ambiguous, LA would increase to 93.4%.

The identification of postmodifying attributes is a hard task for MaltParser, and it found less than 50% of them. Also, parsing the verb chain is quite challenging: only 63% on finite main verbs and 55% of infinite main verbs were recognized. In addition, the detection rate for objects and predicatives could be better (during our experiments,

we observed the rates of 76% and 74%, respectively). However, the subject was properly recognized on 83% of the cases. Also, the parser performs well on premodifying attributes and noun heads which depend on the adposition or quantor heads.

4. Combination of MaltParser and Rule-based Parser

Comparison of error statistics of MaltParser and EstCGParser inspired us to change some labels in MaltParser output to the more reliable EstCGParser's labels. Experiments have shown that verb groups have been analysed better by EstCG. Using the verb group annotation from EstCGParser's output and the rest of the annotation from MaltParser output yielded improvement of LAS by 0.2% and LA by 0.4%. For example, the phrasal verb identification rules are very lexicalized in EstCG and therefore perform with high precision [4].

We also tried to merge annotations of both parsers, and took unambiguous part from the output of EstCGParser and supplemented it with the annotation from output of MaltParser. The LA score remained approximately the same. Probably the most complicated parts of the sentences are hard to analyze for both parsers.

We also had a hypothesis that supplying some information from rule-based parser as input for MaltParser would provide further performance gains.

Our first set of experiments involved clause boundary markup information. Currently, EstCG has ca 80 hand-crafted rules for detecting clause boundaries. Including this automatic annotation in the feature set improved the performance of MaltParser moderately (LAS 80.5, LA 88.7, UAS 83.7), affecting mostly the unlabeled attachment score. The analysis of statistics about specific dependency relations indicated that the performance improvement was homogeneous.

During our next set of experiments we included also labels of syntactic functions to the feature column of the MaltParser input. These functions were found by EstCGParser automatically, and in order to tackle ambiguous analysis, a simple disambiguation heuristic was employed – the first label in the row was selected. Labels attained this way were added to the feature set column of each row in the training and test corpora.

The performance of the combination EstCGparser + MaltParser improved by almost 1%: LAS 81.2, LA 89.6, UAS 84.0.

In our next experiment we have applied a special set of VISL CG-3 rules on top of MaltParser output (transformed to CG format). We used slightly different training and test corpora for this experiment but the training and test corpora were strictly separated.

The initial unlabeled attachment error rate in MaltParser output was 16.2%. Most numerous sources of errors are determining the governor of an adverbial (29% of mistakes) and a nominal attribute (16% of mistakes). 22% of mistakes were caused by incorrect analysis of verbal chains, among those the most numerous were errors in determining the governor of the finite main verb (13.6% of mistakes).

Assuming that the correct relations between the components of a verbal chain (i.e. choosing the correct predicate for the clause) is essential for the correct syntactic analysis of a clause as a whole, the initial post-MaltParser rule-set focuses on fixing these relations and also the relations between clauses as they are overtly expressed as the relations between main verbs of those clauses. It had also to be checked whether the root had only one dependent - mainly a verbal nucleus, but in the case there is no verbal predicate

in the sentence, the first dependent could also be subject or adverbial. For checking and correcting those relations a post-MaltParser rule-set containing 200 rules in VISL CG-3 format was developed.

Applying the post-MaltParser rule-set reduced the amount of clauses having more than one root dependent from 12.6% to 2.6%. Rules reconsidering the function labels and dependency links of the members of verbal chain fixed 25% of the errors concerning the analysis of the verbal chain and one third of the errors concerning dependency links of the finite and infinite main verbs. Another important group of errors were the dependency links of subjects; these were also reduced by one third.

A typical error in the MaltParser output was the wrong attachment of relative clauses. They, more precisely their verbal nucleus, should be attached to the noun they are modifying, but MaltParser often attached them to the verbal nucleus of the main clause, a mistake that was mostly quite easy to fix by the rules.

In the following example (1) the relative clause *kes esimesena kasutab* ... should be attached to the preceding pronoun *see* 'this', but MaltParser has attached it to the governing verb of the main clause *on* 'is'.

(1) Süüdi on see, kes esimesena kasutab füüsilist vägivalda
guilty is this who first uses physical violence
'This (person) is guilty who uses physical violence first.'

The post-MaltParser ruleset for fixing the errors concerning labels and dependency links of the verbal chain reduced the error rate by 1.5%.

5. Conclusions

This paper has provided an overview of the latest developments in computational syntactic analysis of Estonian. First, we presented Estonian Dependency Treebank – an ongoing corpus annotation project with a goal to construct a dependency treebank of Estonian that is annotated for morphological information, syntactic functions and dependency relations. The planned size of the corpus is 400,000 words; all texts are going to be double-annotated and discrepancies resolved by a third, super-annotator. The completion of the treebank is scheduled for the end of 2014.

Although the treebank construction is still under way, we have used it for training MaltParser and for experimenting with combining MaltParser with a rule-based Constraint Grammar parser. We converted the CG annotations to CoNLL format, whereas the syntactic annotations remained the same. We trained the model using Covington's non-projective algorithm, with the size of the training corpus being 153,471 tokens.

MaltParser achieved the labeled attachment score of 80.3%, unlabeled attachment score of 83.4% and label accuracy of 88.6%. Combining the MaltParser with CGparser in different ways improved the LAS by up to 1.5%.

Comparing the outputs of MaltParser and CG parser revealed that CG parser was more efficient when analyzing the verbal chain and determining the clause boundaries, the most the most complex parts of sentences remained challenging for both parsers.

During the error analysis process we found that the training and test corpora contained some errors and inconsistencies in the annotation. Resolving these inconsistencies would improve the performance of our prototype. Likewise, increasing the size of

the training corpus would be equally beneficial for performance. Finally, we are also considering the inclusion of some specific rules to our EstCG grammar: the rules for parsing different quantors, abbreviations, direct speech, and bibliographical references in scientific texts.

During our future work, we plan to address aforementioned issues and apply presented methods to automatically morphologically analyzed and disambiguated input.

Acknowledgements

This work was supported by Estonian Ministry of Education and Research (grant IUT20-56 "Eesti keele arvutimudelid / Computational models for Estonian") and the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science (EXCS).

References

[1] K. Müürisep, T. Puolakainen, K. Muischnek, M. Koit, T. Roosmaa, and H. Uibo. A New Language for Constraint Grammar: Estonian. *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2003.* Borovets, Bulgaria, 2003. pp. 304–310.

[2] F. Karlsson, A. Anttila, J. Heikkilä, and A. Voutilainen. *Constraint Grammar: a Language-Independent System for Parsing Unrestricted Text.* Mouton de Gruyter. 1995.

[3] E. Bick. *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework.* Aarhus University Press, Aarhus. 2000.

[4] K. Muischnek, K. Müürisep, and T. Puolakainen. Estonian particle verbs and their syntactic analysis. *In Human Language Technologies as a Challenge for Computer Science and Linguistics: 6th Language & Technology Conference Proceedings.* (Eds.) Zygmunt Vetulani and Hans Uszkoreit. Poznan 2013. pp. 338–342.

[5] J. Nivre, J. Hall, and J. Nilsson. Malt-parser: A data-driven parser-generator for dependency parsing. *In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)* 2006. pp. 2216–2219.

[6] M. Ballesteros and J. Nivre. MaltOptimizer: A System for MaltParser Optimization, *In Proceedings of LREC 2012*, pp. 2757–2763.

[7] J. Nilsson, J. Nivre. MaltEval: An Evaluation and Visualization Tool for Dependency Parsing. *In Proceedings of LREC 2008*, pp. 161–166.

[8] A. Hakulinen, M. Vilkuna, R. Korhonen, V. Koivisto. *Iso suomen kielioppi / Grammar of Finnish.* Suomalaisen kirjallisuuden seura. 2004.

[9] A. Voutilainen, T. Purtonen, and K. Muhonen. *FinnTreeBank2 Manual*. 2012.

[10] K. Haverinen, J. Nyblom, T. Viljanen, V. Laippala, S. Kohonen, A. Missilä, S. Ojala, T. Salakoski, and F. Ginter. Building the essential resources for Finnish: the Turku Dependency Treebank. *Journal of Language Resources and Evaluation* July 2013.

[11] M. de Marneffe and C. Manning. *Stanford typed dependencies manual.* Technical report. Stanford. 2008.

[12] L. Pretkalnina, A. Znotins, L. Rituma, D. Gosko. Dependency parsing representation effects on the accuracy of semantic applications - an example of an inflective language. *In Proceedings of LREC 2014*, pp. 4074–4081.

[13] J. Kapociute-Dzikiene, J. Nivre, and A. Krupavicius. Lithuanian Dependency Parsing with Rich Morphological Features. *In Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages*, 2013. pp. 12-21.

[14] R. T. Mcdonald, J. Nivre, Y. Quirmbach-Brundage, Y. Goldberg, D. Das, K. Ganchev, K. B. Hall, S. Petrov, H. Zhang, O. Täckström, C. Bedini, N. Bertomeu, Castelló, J. Lee. Universal dependency annotation for multilingual parsing. *In Proc. of ACL'13*, 2013 pp. 92–97.