Acta Linguistica Academica Vol. **64** (2017) 3, 347–367
DOI: 10.1556/2062.2017.64.3.2

Parsing and beyond

Tools and resources for Estonian

Kadri Muischnek
University of Tartu, Estonia
kadri.muischnek@ut.ee

Kaili Müürisep
University of Tartu, Estonia
kaili.muurisep@ut.ee

Tiina Puolakainen
Institute of the Estonian Language, Tallinn, Estonia
tiina.puolakainen@eki.ee

Abstract: This article gives an overview of the state of art of tools and resources for syntactic analysis of Estonian. A morphosyntactic disambiguator, surface-syntactic analyzer and dependency parser are all based on the Constraint Grammar formalism. As for language resources, a 400,000-word manually annotated dependency treebank has been created, its annotation scheme is compatible with the output of the Constraint Grammar dependency parser. Part of the treebank has been converted to the Universal Dependencies annotation scheme. Our tools have also been tested by large-scale corpus annotation.

Keywords: morphological disambiguation; dependency parsing; treebank; Constraint Grammar; Universal Dependencies; Estonian

1. Introduction

This paper presents a set of tools and resources for parsing Estonian texts, starting from morphological analysis and disambiguation to dependency parsing. The main purpose of this article is to apprise the potential users of these Estonian language resources and encourage their utilization.

First, some historical background. In 1995 ESTMORF, the first version of a morphological analyzer and guesser of Estonian was created and a couple of years later it was able to assign adequate morphological descriptions to 99% tokens in text (Kaalep 1997; Kaalep & Vaino 2001).

In the same year, Fred Karlsson together with his colleagues published a monograph on Constraint Grammar (Karlsson et al. 1995), a framework for disambiguating and parsing unrestricted text that has been successfully used not only for analyzing the Indo-European languages, but also, for example, for analyzing Finnish. That spurred the work on Estonian Constraint Grammar (EstCG).

Estonian linguistic tradition (e.g., Erelt et al. 1993) is based on traditional descriptive grammar, analyzing the syntactic structure using the notion of syntactic functions, also known as syntactic relations. There are no phrase-structure-based descriptions established on a formal grammar theory, nor (full) models of Estonian syntax. So, Constraint Grammar, which depicts the surface syntactic structure of a sentence using syntactic function labels, seemed convenient for building the first parser of Estonian. Its earlier versions used a locally developed parsing engine (Müürisep 2000), but its last version uses VISL CG-3 format and software (Bick & Didriksen 2015).

Estonian Constraint Grammar parser consists of separate sets of grammar rules for determining clause boundaries, morphological disambiguation, surface syntactic analysis and dependency relations. A grammar rule removes superfluous readings based on the context. The context conditions of rules are similar to regular expressions, they can be linked to any tag or any word anywhere in a sentence, using defined or undefined distances. Context conditions of the same rule may be complex, consisting of smaller conditions, linked together.

The output can remain ambiguous, i.e., a word-form can have more than one label.

In addition to rule-sets, the system also includes several valency lexicons and a special module for identifying particle verbs (Muischnek et al. 2013).

The morphological disambiguator needs an output from a morphological analyzer as its input. There is a free open-source morphological analyzer of Estonian: *Vabamorf*.[1] Its output is converted to CG format, and using the lexicons, additional valency annotation has been included (e.g., transitiveness of verbs, information about possible particle verbs, valency information of pre- and postpositions, etc.) The different modules of Estonian Constraint Grammar are applied like a pipeline, one by one. First, preliminary clause boundary detection rules are applied, then morphological disambiguation takes place, after that particle verbs are recognized and the particle and the verb are linked together. Then, surface syntactic rules are applied, which add a syntactic annotation label to each morphological reading of word forms. The last set of rules establishes dependency links between word forms. The parsing system can be invoked as a command

[1] https://github.com/Filosoft/vabamorf

line pipeline, but it is also available as a Python module (Orasmaa et al. 2016).[2]

The rest of the paper expands on the aforementioned modules and is organized as follows: sections 2 and 3 provide an overview of morphological disambiguation and clause boundary detection modules. Sections 4 and 5 describe the grammar of surface and dependency syntax. Section 6 reports the experimental results of applying MaltParser to Estonian. In section 7 we introduce Estonian treebanks. Section 8 presents some syntax-based applications followed by a conclusion and some ideas for future work.

2. Morphosyntactic disambiguator

The EstCG parser takes morphologically analyzed text as input, i.e., each word-form has all the possible morphological analyses attached to it. According to Tiina Puolakainen, the morphological ambiguity rate of an Estonian text is about 45% (Puolakainen 2001, 14). Earlier experiments with Corpus of Estonian Literary Language have yielded similar ambiguity rate (Kaalep 1997). For example, the word-form *või* can be a noun *või* ‘butter’ in nominative or genitive case-form, or a negative present tense form of the verb *võima* ‘may’ in all three persons in singular and plural or conjunction *või* ‘or’. The word form and all its morphological interpretations constitute a cohort and these interpretations are called readings.

Figure 1 illustrates morphological ambiguities of Estonian sentences: although morphological analyzer and guesser analyze about 99% of words correctly, the average number of readings per word is about 1.4 (punctuation marks have not been taken into account when calculating this figure).

Constraint Grammar rules for morphological disambiguation delete readings that are inappropriate regarding the context, one by one. If it is not possible to disambiguate based on the contextual information, all possible readings are retained.

The disambiguating grammar consists of more than 3,400 handwritten rules that tend to be quite specific: almost a quarter of them address single word-forms. But of course the grammar also contains more general rules covering broader ambiguity classes. For example, the choice between readings of adposition and adverb is based on rules which check whether there exists a suitable candidate for the noun that should be governed by the adposition, i.e., a noun that is in the appropriate morphological case. Also, if an ambiguous noun is part of an adpositional phrase, the

[2] https://github.com/estnltk

```
"<Inimesed>"                                                %people
      "inimene" Ld S com pl nom cap
"<on>"                                                      %have
      "ole" L0 V aux indic pres ps3 pl ps af <Intr>
      "ole" L0 V main indic pres ps3 sg ps af <Intr>
      "ole" L0 V aux indic pres ps3 sg ps af <Intr>
      "ole" L0 V main indic pres ps3 pl ps af <Intr>
"<linna>"                                                   %city
      "linn" L0 S com sg gen
      "linn" L0 S com sg adit
      "linn" L0 S com sg part
"<metsa>"                                                   %forest
      "mets" L0 S com sg adit
      "mets" L0 S com sg part
      "mets" L0 S com sg gen
"<rajanud>"                                                 %build
      "raja" Lnud V main partic past ps
      "raja" Lnud V main indic impf ps neg
      "raja=nud" L0 A pos partic <nud>
      "raja=nu" Ld S com pl nom <nu>
"<.>"
      "." Z Fst
```

Figure 1: The sentence *Inimesed on metsa linna rajanud* 'People have built a city in the forest' prior to morphological disambiguation.

government of the adposition can be used to determine the case of the word form.

While the Constraint Grammar rules of surface syntax can be seen as a simplified formalization of the descriptive grammar of Estonian (intransitive verbs do not have objects, a clause may only have one uncoordinated subject, etc.) then it is very difficult to give an abstract description to morphological disambiguation rules. They try to choose the correct reading for a word-form based on unambiguous context.

A difficult case for disambiguation is the choice between the readings of nominative, genitive, partitive, and short illative (also termed as aditive) case forms of a noun. This type of ambiguity tends to be more characteristic of frequent and common words, e.g., noun forms *ema* 'mother' and *isa* 'father' are ambiguous between nominative, genitive, and partitive readings.

These ambiguities are illustrated in Figure 1, depicting the sentence (1) prior to the morphological disambiguation process. The word-form *metsa* is an example of typical word-form homonymy, as it can be the word *mets* ‘forest’ in singular genitive, partitive, or aditive (short illative) case, whereas, as is often is the case, the parallel regular form of illative case – in this example *metsasse* ‘into the forest’ – although present in the morphological paradigm, is not in actual use.[3]

(1) Inimesed on metsa linna rajanud.
human-PL are forest-ADIT city-GEN found-PS.PTCP
‘People have built a city in the forest.’

Note that the sentence is also semantically ambiguous: both readings - ‘People have built a city in the forest’ and ‘People have built a forest in the city’ are equally possible as both word-forms *linna* and *metsa* have the possible readings of the object cases (genitive and partitive) and the readings of aditive or short illative. The word order of Estonian is free, so one cannot rely on that for disambiguation. Also, the grammatical (though nonsensical) reading ‘People are the ones who built a forest of a city’ is also possible. In this case, both *linna* and *metsa* are in the genitive case, and *rajanud* is a noun in plural nominative.

Other frequent sources of morphological ambiguity are past participles and word-forms ambiguous between the readings of the adposition, adverb, and inflectional form of a noun or verb.

Due to the grammatization process, several adpositions and adverbs in Estonian have emerged from inflectional forms of nouns or verbs. For example, the word-form *peale* can be an autonomous adverb (the most general meaning of which is ‘onto’) or a particle as a part of a particle verb, e.g., *peale sattuma* ‘stumble on/across’. It can also be a postposition governing a noun in the genitive case (meaning ‘in addition to’), or elative case (meaning ‘starting from’); or preposition governing a noun in the genitive case (mostly equivalent to an allative case form of the noun), or partitive case (meaning ‘after’): after all, or diachronically, before all: *peale* can also be also a noun, *pea* meaning ‘head’, in a singular allative case. These possible analyses are depicted as examples (2)–(5):

(2) Laps tegi oma peale haiget.
Child did own head-ALL pain-PRT
‘Child hurt her/his head.’

[3] List of abbreviations used in the glosses: ADIT = aditive, short form of illative case, ALL = allative case, COND = conditional, COM = comitative, GEN = genitive case, INE = inessive case, INF = infinitive, PL = plural, PRT = partitive case, PST = past, PTCP = participle, SG = singular.

(3) Film ajas une peale.
Movie drove sleep-GEN onto
'The movie made someone sleepy.'

(4) Pani lina laua peale.
Put tablecloth-GEN table-GEN onto
'S/he put a tablecloth on the table.'

(5) Peale tööd lähen koju.
After work-PRT go-1.SG home
'I will go home after work.'

Another difficult case for disambiguation are past participles, which are always ambiguous in four ways, although the fourth possible analysis is very infrequent. The more possible readings are those of negative indicative past tense, past participle, and adjectival use of past participle. In addition to that, past participle, as a nominalization of its adjectival reading, can act as noun in plural.

For evaluating the parsing modules, we have used the corpus of 25,719 tokens (19,015 words without punctuation marks) that consisted of fiction, scientific and newspaper texts (see Table 2 for details).

Morphological analyzer and guesser were able to find correct readings for 99.18% of words, with each word having on average 1.4 readings (42–49% words had ambiguities). Most ambiguities occurred when analyzing names. We also observed a number of unknown words in scientific texts that were not present in lexicon and were thus hard to analyze (e.g., *ellipsoid* 'ellipsoid'). After disambiguation 96.84% of words had a correct morphological analysis, while 5% of words remained ambiguous (recall[4] 96.84%, precision[5] 94.1%).

The disambiguating rules, of course, make use of sentential context, and especially information about the finite verb form in the clause. In order to do so, the clause boundaries need to be known.

One of the most difficult tasks is disambiguating noun forms with homonymous nominative, genitive, and partitive; or genitive, partitive, and aditive case forms, both of them are frequent forms of ambiguity, e.g., word-forms *linna* and *metsa* on Figure 1.

[4] Recall is the ratio of the correct analyses in the output to the analyses in the golden standard.

[5] Precision is the ratio of the correct analyses in the output to all analyses in the output.

Even if the clause boundaries are recognized correctly, still a common source of errors are verbless clauses: elliptical sentences and sentence fragments, often used as headlines. The rules are designed so that they "aim at" finding a finite verb form for every clause. For example, sentence (6) contains three word-forms, two of them are three-way and one two-way ambiguous. The nouns *vaba* 'free') and *raha* 'money' can be nominative, genitive, or partitive case forms. *Puudus* 'lack' can be a noun in nominative case form, or a singular 3rd person past tense form of the verb *puuduma* 'lack'. As the latter is the only candidate for a finite verb in this clause, it is erroneously disambiguated as a verb form.

(6) Vaba raha puudus.
Vacant-GEN money-GEN lack
'Lack of vacant money.'

Errors by morphological analyzer can also lead to additional morphological disambiguation errors, since invalid context facilitates incorrect decisions. For example, in sentence (7) the allative case form of a proper noun *Ram* was incorrectly analyzed as the genitive case form of a proper noun *Ramil*, and therefore the following word form *pähe* (pea 'head' in aditive case) was erroneously annotated as postpositition.

(7) Ta koputas kepiga Ramile pähe.
he knocked stick-COM Ram-ALL head-ADIT
*he knocked stick-COM Ramil-GEN POSTPOSITION
'He knocked Ram with the stick to the head.'

3. Clause boundary detector

Clause boundary annotation is a simple way to constrain the context of morphosyntactic disambiguation rules; doing syntactic analysis is of course impossible without knowing clause boundaries. Also performance of statistical parser improved if the model had information about clause boundaries (Muischnek et al. 2014a).

Currently, the EstCG contains approximately 80 hand-crafted rules for detecting clause boundaries, considering mainly conjunctions, punctuation marks, finite verbs, relative adverbs, and pronouns. Although these are simple cues for assuming a clause boundary, often it is not obvious how to distinguish coordinated clauses from other types of coordination, as a morphologically analyzed (but not yet disambiguated) text contains plenty of ambiguities for different interpretations.

The following example illustrates how punctuation marks get annotated with a CLB label, if both the left and right contexts contain a finite verb before conjunction words and punctuation marks:

```
MAP (CLB) TARGET (Z) (*-1 FinV BARRIER Conj|Pnct)(*1 FinV BARRIER Conj|Pnct);
```

Special clause boundary tags are introduced for embedded clauses, where, for example, a subject and a predicate of the main clause may be separated by a relative clause and, therefore, would not be linked to each other without special effort.

4. Surface-oriented syntactic analyzer

The surface-syntactic module of Estonian Constraint Grammar adds a syntactic function label to every word-form in the text. The repertoire of syntactic functions is based on Constraint Grammar (Karlsson et al. 1995) and a descriptive grammar of Estonian (Erelt et al. 1993) and has been presented in Table 1.

The annotation created by the analyzer is shallow: the clause boundaries are set and the syntactic functions of the word-forms in every clause are labelled, but no syntactic structure (phrases or dependency relations) is determined. It is the task for the next module, which applies after determining syntactic functions.

According to the Estonian Constraint Grammar annotation scheme, members of the verbal chain can be finite or infinite main verbs (FMV, IMV), and finite or infinite auxiliaries (FCV, ICV). A small closed class of verbs including *olema* 'be' in compound tense forms and modal verbs in modal constructions is annotated as auxiliaries. Other finite components of verb clusters, e.g., inchoative verbs like *hakkama* 'to start, to begin' are labelled as main verbs; it means that a verbal chain can consist of two main verbs, one of them finite and the other infinite.

We also distinguish adverbial particles as parts of particle verb (VPart), and verb negators (NEG). Particle verbs are a frequent phenomenon in Estonian, e.g., the example sentence *Hommikul püüdis kass kinni kena paksu hiire* 'In the morning, the cat caught a nice fat mouse' in Figure 3 contains a particle verb *kinni püüdma* 'to catch', literally 'to catch down'.

The arguments of the verb are labelled as subject (SUBJ) object (OBJ), predicative (PRD), or adverbial (ADVL). The attributes of a nominal are tagged according to their part-of-speech (AN for adjectival attributes, NN for nominal, KN for adpositional, DN for adverbial, and INFN for infinitival attributes).

There is no direct connection between an attribute and its head on this level, but pre- and post-modifying attributes are distinguished: there is a special symbol indicating whether the word-form is a pre- or post-modifier (<NN or NN>, for example).

Also, we label direct addresses (VOC), conjunctions (J), and interjections (I). Direct addresses, of course, are rare in more formal written texts, but used more in the texts of social media.

The adverbials (ADVL) form a large and heterogeneous class. Furthermore, sentence and phrase adverbials are not distinguished, so both word-forms *väga* 'very' and *kiiresti* 'quickly' get the label ADVL in the sentence (8).

(8) Ta jooksis väga kiiresti.
s/he (SUBJ) ran (FMV) very (ADVL) quickly (ADVL)
'S/he ran very quickly.'

For adposition and quantifier constructions, there are two theoretical possibilities: one can analyze adposition or quantifier as a modifier of a noun, or label these nouns as modifiers of adpositions or quantifiers. Following the descriptive grammar of Estonian (Erelt et al. 1993), the second solution is chosen in Estonian Constraint Grammar, so nouns governed by an adposition are annotated with a special label (<P or P>) and also nouns governed by a quantifier (<Q or Q>). The adposition or quantifier is labelled according to the syntactic function of the adposition or quantifier phrase as a whole. So in sentence (9), the postposition *taga* 'behind' is labelled as adverbial and the quantifier *palju* 'many, much' as subject.

(9) Akna (P>) taga (ADVL) on (FMV) palju (SUBJ) sääski (<Q)
window-GEN behind is many/much mosquito-PL.PRT
'There are a lot of mosquitoes behind the window.'

The summary of all labels has been provided in Table 1.

Again, following the main grammatical description of Estonian (Erelt et al. 1993), only finite clauses are regarded as clauses. Also, the head verbs are not connected with their arguments in any way. For example, if a clause contains an infinitival subclause and both verbs, finite and infinite, have an object, there is no way to tell from the annotation which object complements which verb. For example, the sentence (10) contains two uncoordinated objects: *teda* 's/he' in partitive case and *šokolaadi* 'chocolate' in partitive case, and two transitive verb forms: *haaras* 'captured' and *süüa*

Table 1: Labels of syntactic functions

Label	Explanation
FMV	Finite main verb
FCV	Finite chain verb
IMV	Infinite main verb
ICV	Infinite chain verb
NEG	Negation
Vpart	Verbal particle
SUBJ	Subject
OBJ	Object
PRD	Subject complement
ADVL	Adverbial
NN> <NN	Nouns as pre- and postmodifiers
AN> <AN	Adjectives as pre- and postmodifiers
KN> <KN	Adpositions as pre- and postmodifiers
DN> <DN	Adverbs as pre- and mostmodifiers
Q> <Q	Complements of quantors
P> <P	Complements of post- and premodifiers
VOC	Direct address
J	Conjunction
I	Interjection
???	Unknown syntactic function

'to eat', but there is no way to tell from the surface syntactic annotation, which verb is governing which object.

(10) Teda (OBJ) haaras (FMV) vastupandamatu (AN>) soov (SUBJ)
s/he-PRT capture-3.SG.PST irresistible wish
šokolaadi (OBJ) süüa (<INFN)
chocolate-PRT eat-INF
'An irresistible wish to eat chocolate captured him/her.'

These kind of syntactic ambiguities partly motivated further deeper syntactic analysis. This is the goal of the next EstCG grammar module, a module for building dependency trees described in section 5.

The rule-set for determining the syntactic function labels comprises approx. 1,300 rules. First, all possible labels are added depending on the part-of-speech tag and grammatical categories present in word-form. Then, the syntactic labels that do not conform with other labels, or morphological information present in the same clause, are deleted, one by one.

For example, a noun in partitive case form gets the label of direct object during the initial mapping phase, but it also gets several other syntactic labels. The object label is deleted, if the finite verb in that clause is an intransitive one, or it is a verb that under certain circumstances takes only a total object[6] (i.e., an object in genitive or nominative case), or, if the same clause contains a noun with a non-ambiguous object reading, and the word-form under consideration is not in a coordinating relation with it. The following rule removes objects' labels if there is only one verb in the clause, there is also an unambiguous object in the clause, and there are no punctuation marks or conjuncts between the current word and the object:

```
REMOVE (@OBJ) (NOT 0 Inf) (NEGATE *0 ParticSupInfGer BARRIER CLB)
               (*0C Objekt BARRIER Conj|Pnct OR CLB) ;
```

Example (11) contains two nouns in partitive case: *jootraha* 'tip' and *mõte* 'sense, thought'. The sentence contains one transitive verb form, *anda* 'to give', and one intransitive verb form, *pole* 'is not'. Both nouns are possible objects of *anda* and the correct one (*jootraha*) is actually separated from the governing verb by the other possible candidate *mõtet*.

(11) Seega pole jootraha (OBJ) mõtet (SUBJ) anda.
thus is-not tip-PRT sense-PRT give-INF
'Thus tipping does not make sense.'

Experiments on the same automatically analysed corpus showed that the recall of the syntactic analysis was 90.8% and precision 80.8% (see Table 2).

Here, recall is defined as the ratio 'assigned appropriate labels/all appropriate labels', and precision as the ratio 'assigned appropriate labels/all assigned labels'. It means that 9.2% of tokens do not get the correct label, and 19.20% of the added labels are either superfluous, or erroneous. As the surfaces syntax rules have been developed on the fiction texts mainly, the results on that genre are significantly better: recall 93% and precision 85.6%. The weakest parts of the surface syntax rules are the analyses of sentences with digital numbers (as they have no morphological information) or multiword foreign names.

[6] Grammatical aspect in Estonian has not developed into a consistent grammatical category, but it emerges in the object case alternation. One can read about the complicated system of Estonian object case alternation in (Erelt 2003, 96–97).

Table 2: Performance of rule-based parser[7]

	Tokens	Words	Recall of morph. anal.	Recall of disambiguator	Precision of disambiguator	Recall of surface syntax analyzer	Precision of surface syntax analyzer	Rule-based UAS
Fiction	11467	7979	99.50	97.37	94.38	93.0	85.6	79.8
Science	11217	8745	98.94	96.55	94.16	89.4	77.9	75.7
Newspapers	3026	2291	98.69	96.11	92.91	88.5	76.1	74.8
Total	25710	19015	99.18	96.84	94.1	90.8	80.8	77.3

The majority of errors occur in annotating objects, subjects, and predicatives (subject complements), as they can be coded using the same morphological cases. A noun in nominative case form can be a subject, an object, or a predicative. A noun in genitive case form can be an object (only in singular), or a genitive attribute. A noun in partitive case form can be a subject, object, predicative, or a modifier of a quantifier. Also, the nouns in nominative, genitive, or partitive case that can act as adverbials (of time and measurement) belong to the adposition phrase, or perform some less observed role in the sentence.

Again, the sentence *Inimesed olid metsa linna rajanud* in Figure 1 has two word-forms – *linna* and *metsa* – that are morphologically three-way ambiguous (the readings on singular genitive, partitive, and aditive case forms) and an error, or remaining ambiguity on the morphological level creates an error, or multiplies the ambiguity on the syntactic level.

Examples (12)–(13) illustrates the syntactic ambiguity caused by a word-form in the genitive case. Both sentences (12) and (13) contain the word-form *etenduse* ‘show’ in singular genitive case, followed by a noun in inessive case form. In sentence (12) it is an attributive noun, governed by adverbial *lõpus* ‘end’ in singular inessive case form. In sentence (13) the word-form *etenduse* is the object of the main verb *teeme* ‘do’ in 3rd person plural and, just like in the previous example, is followed by a noun in singular inessive case form.

(12) ... tulevad lavale etenduse (OBJ **NN>**) lõpus
come-3.PL stage-ALL show-GEN end-INE
‘... they come to the stage at the end of the show’

[7] The files are: ilu_kivirahk.tasak.inforem, tea_geofyysika_10000.tasak.inforem, and aja_EPL_2007_08_12.tasak.inforem, available at https://tinyurl.com/y83c3g7u.

(13) ...teeme vähemalt ühe etenduse (**OBJ** NN>) kuus
do-1.PL at-least one-GEN show-GEN month-INE
'... we give at least one show per month'

As pointed out in Section 2, disambiguating between the readings of nominative, genitive, partitive, and aditive (short illative) case forms is a hard task for the morphological disambiguator. In case the morphological disambiguation fails, the syntactic analyzer faces a situation where there are several multi-way ambiguous nouns in the clause, and in this situation the syntactic analysis often fails.

A substantial amount of non-solved ambiguity in the output is caused also by the indiscernibility of adverbials and adverbial attributes. The problem is similar to pp-attachment. In example (14), a noun in inessive case, *puusärgis* 'coffin', modifies the main verb, *pääsema* 'access', as an adverbial, but in example (15) a noun in inessive case, *raudrüüs* 'armor', modifies the noun, *rüütel* 'knight', as an attribute.

(14) Kui pääseksin oma puusärgis (**ADVL** NN>) paradiisi
if access-COND.1.SG own coffin-INE paradise-ADIT
'If I could get to paradise in my own coffin.'

(15) Teed mööda tuli raudrüüs (ADVL **NN>**) rüütel
Road-PRT along came armor-INE knight
'Along the road came an armored knight.'

5. Dependency parser and particle verb detector

Recently, the EstCG parser has been enhanced with dependency rules; this stage is still under development (Muischnek et al. 2014a). The coding of dependency relations is based on an expansion of Constraint Grammar (Bick & Didriksen 2015). However, the analysis provided by CG dependency parser helped to develop the first version of the Estonian Dependency Treebank, consisting of 400,000 words (Muischnek et al. 2014b), which in turn gave an opportunity to experiment with statistical parsing methods, namely training and evaluating MaltParser (Nivre et al. 2007) for analysing Estonian texts.

The grammar of dependencies consists of approx. 600 rules. Several rules consider context over clause boundaries in order to bond the surface-syntactically annotated sentence to a syntax tree. The EstCG parser achieves an unlabeled attachment score (UAS) of 77.3% (see Table 2).

We added a special module of rules in order to recognize particle verbs, i.e., multi-word expressions consisting of a verb and an adverbial particle, also called phrasal verbs in more general terms. Particle verbs are a frequent phenomenon in Estonian, and their formation is partly a productive process: one can add the perfective particle *ära* to almost every verb. As it is the case also in some other languages (cf. e.g., Villavicencio & Copestake 2002), one can join particle verbs by combining nearly all verbs of movement with directional particles. In Estonian, they have been claimed to be a structural loan from German (Hasselblatt 1990) and indeed, a great part of Estonian particle verbs have direct counterpart in German, also their sentence distribution reminds that of German particle verbs.

The module for identifying particle verbs consists of approx. 500 rules and a thorough lexicon containing lists of particles, and corresponding lists of verbs.

As our results indicate, our lexicon- and rule-based approach can be regarded as successful. More than 95% of the particle verbs receive correct analysis at the shallow syntactic level, and 95–100% of the particle verbs get correct dependency relations (i.e., the particles get combined with correct verbs), what makes it possible to use annotated data for practical linguistic purposes. The module is described in more detail in (Muischnek et al. 2013).

6. Statistical parser

For our first experiments we have selected MaltParser (Nivre et al. 2007), since it has been successfully employed for a wide range of languages, including languages with inflectional morphology and relatively small treebanks (for example, Latvian and Lithuanian). In addition, MaltParser includes the MaltOptimizer system (Ballesteros & Nivre 2014) which helps the end user to select the appropriate parameters and parsing algorithm without having expert knowledge on underlying methods.

First, we transformed the corpus texts from CG format to CoNNL-X format. As the regular set of POS tags consists of 15 tags, there is also an option to employ 22 fine-grained POS tags. Most of morphological description has been retained except valency information (intransitivity of verb or possible object cases for transitive verb). The syntactic labels remain the same as in the EstCG 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, marking only that there exists a dependency relation to the main or coordinated clause. The root of the clause gets only one label, that indicates its function in the context of this clause. If we wanted to annotate the syntactic function of this clause in respect of the governing clause, we would have to introduce secondary edges, or express only this information. The latter option has been chosen in Universal Dependencies', for example.

Only a part of the treebank that was double-checked (191,000 tokens, 13,310 sentences) was used for statistical parsing. 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 consisted of 153,471 tokens. First, we used MaltOptimizer to find the most appropriate training model and parameters. The tool suggested using a Covington-Non-Projective algorithm and a specific feature model.

The preliminary results gave labelled 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 the label NONE.

After excluding punctuation marks and non-sentential constructions from the evaluation, 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%.

We have conducted several experiments running the Maltparser along with the EstCG parser: using syntactic information provided by the EstCG parser as input for Maltparser or applying special fixing rules to the output of Maltparser. These improved overall performance by 1% (Muischnek et al. 2014a).

The preliminary tests on automatically morphologically annotated data yielded by approx 2% lower UAS, but Maltparser still performs significantly better on building dependency links.

7. Corpora and treebanks

As already mentioned, the initial versions of the EstCG parser were developed based on linguistic knowledge as presented in a descriptive grammar of Estonian (Erelt et al. 1993), and a small experimental test and development corpus (12,000 words). In order to improve the coverage of the rule-based CG parser, and to experiment with machine learning based parsers, creating a larger manually annotated corpus was essential. We succeeded in getting funding for the creation of an Estonian Dependency Treebank, and completed its first version by the end of 2014 (Muischnek et al. 2014b). The treebank contains approximately 400,000 tokens and is annotated for part of speech, morphological description, syntactic functions and dependency relations.[8]

Figure 2 depicts the Dependency Constraint Grammar analysis of an Estonian sentence, *Hommikul püüdis kass kinni kena paksu hiire* 'In the morning, the cat caught a nice fat mouse', as it appears in the Estonian Dependency Treebank.

For every word in the sentence there is a separate row containing its analysis. It begins with lemma, e.g., *hommik* 'morning) for the word-form *hommikul* 'in the morning' in the singular adessive case. Lemma is followed by an inflectional ending, a string beginning always with L; e.g., Ll for the word-form *hommikul.* Then comes the POS tag, e.g., S (noun) for *hommikul* 'in the morning' or V (verb) for *püüdis* 'caught', followed by morphological description. The syntactic function labels begin with @ and tags indicating dependency relations with #.

Figure 3 presents a visualization of the same tree, generated with *brat* software.[9]

In order to join in an international effort to make the Estonian Dependency Treebank available with a cross-linguistically consistent treebank annotation, we have started converting the aforementioned treebank to the Universal Dependencies (McDonald et al. 2013) annotation scheme.[10]

Part (approx. 224,000 tokens) of EDT was converted to UD format and released in Universal Dependencies version 1.3 in May 2016 (Muischnek et al. 2016). The Estonian UD treebank available via Universal Dependencies website is a little larger, approx. 234,000 tokens, as it also contains sentences from another smallish phrase-structure treebank that was automatically converted to UD format (Rosa et al. 2014).

In the near future, we hope to convert also the remaining part of the EDT to UD format.

Figure 4 presents UD-style tree for the sentence 'In the morning, the cat caught a nice fat mouse'.

[8] It is freely available from https://github.com/EstSyntax/EDT.

[9] http://brat.nlplab.org/

[10] https://github.com/EstSyntax/EstUD

```
"<s>"
"<Hommikul>"                                            %morning
      "hommik" Ll S com sg ad cap @ADVL #1->2
"<püüdis>"                                              %caught
      "püüd" Lis V main indic impf ps3 sg ps af @FMV #2->0
"<kass>"                                                %cat
      "kass" L0 S com sg nom @SUBJ #3->2
"<kinni>"                                               %verbal particle
      "kinni" L0 D @Vpart #4->2
"<kena>"                                                %nice
      "kena" L0 A pos sg gen @AN> #5->7
"<paksu>"                                               %fat
      "paks" L0 A pos sg gen @AN> #6->7
"<hiire>"                                               %mouse
      "hiir" L0 S com sg gen @OBJ #7->2
"<.>"
      "." Z Fst CLB #8->8
“</s>"
```

Figure 2: Full analysis for the sentence

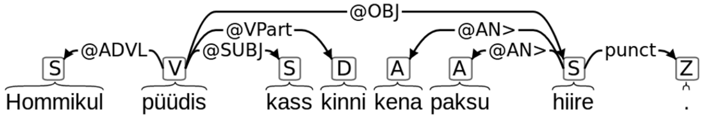

Figure 3: Visualization of dependency analysis for the sentence ‘In the morning, the cat caught a nice fat mouse’

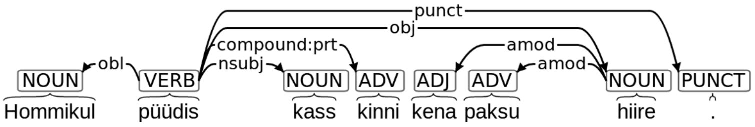

Figure 4: UD-style dependency analysis for the sentence ‘In the morning, the cat caught a nice fat mouse’

The CG dependency parser has also been used for large-scale corpus parsing. A parsed version of *Tasakaalus Korpus*, the Balanced Corpus of Estonian[11] is available for querying via the portal *Keeleveeb*.[12]

The original EstCG syntactic analyzer has been created for standard written Estonian, but the modules are easily adaptable for analyzing other language varieties. Experiments have been made with special extra rule-sets for spoken Estonian (Müürisep & Nigol 2008; 2009), Estonian as used in social media (Särg 2015), and also for analyzing Estonian dialects (Lindström & Müürisep 2009).

8. Conclusions and future work

Our EstCG is a rule-based model. The advantages of a rule-based model include adaptability, and the possibility of gradual improvement and refinement of the rule-sets.

As for the main disadvantages, one could name the large number of rules that, if not properly organized, can make the rule-set unmanageable. A special effort is needed in order to keep the rule-set consistent.

The main shortcoming of EstCG tagset is perhaps the high granularity of morphological tagset, as this feature makes it difficult to combine EstCG with statistical morphological disambiguator.

Building a morphosyntactic and syntactic analyzer, or parser, can be an interesting task, per se, and building large syntactically annotated corpora promotes both language technology and linguistic research. But, of course, our aim is also to foster using Estonian Constraint Grammar in applications. Among those one could mention language learning programs Oahpa! and Vasta!, developed at Giellatekno (Antonsen et al. 2009a;b); programs using linguistic tools for generating new tasks for language learners and testing the students' answer, enabling more flexibility for the generated tasks and the possible answers, and more deliberate and precise feedback to the student accordingly to particular linguistic issues relevant for a student's answer. Estonian Oahpa! (Uibo et al. 2015) and Vasta! are currently under development.

Another system in which we are planning to employ Estonian Constraint Grammar is rule-based machine translation platform Apertium (Forcada et al. 2011).

[11] https://tinyurl.com/yddh3yyv

[12] http://www.keeleveeb.ee/

The reader can test our demo version of the syntactic parser on a website[13] or install it as an open-source software.[14] In addition to original EstCG modules, one can use EstNLTK (Orasmaa et al. 2016), an open source Python toolkit for analyzing Estonian texts.

Acknowledgements

This work was supported by the Estonian Ministry of Education and Research (grant IUT20-56 ”Eesti keele arvutimudelid/Computational models for Estonian”) and the National Programme for Estonian Language Technology.

References

Antonsen, Lene, Saara Huhmarniemi and Trond Trosterud. 2009a. Constraint grammar in dialogue systems. In Proceedings of Workshop on Constraint Grammar and robust parsing at NODALIDA 2009 (NEALT Proceedings 8). 13–21.

Antonsen, Lene, Saara Huhmarniemi and Trond Trosterud. 2009b. Interactive pedagogical programs based on constraint grammar. In Proceedings of the 17th Nordic Conference of Computational Linguistics NODALIDA (NEALT Proceedings 4). 10–17.

Bick, Eckhardt and Tino Didriksen. 2015. CG3 Beyond classical constraint grammar. In Proceedings of the 20th Nordic Conference of Computational Linguistics NODALIDA 2015. Linköping: Linköping University Electronic Press. 31–40.

Ballesteros, Miguel and Joakim Nivre. 2014. MaltOptimizer: Fast and effective parser optimization. Natural Language Engineering 22. 187–213.

Erelt, Mati. 2003. Estonian language. Tallinn: Estonian Academy Publishers.

Erelt, Mati, Reet Kasik, Helle Metslang, Henno Rajandi, Kristiina Ross, Henn Saari, Kaja Tael and Silvi Vare. 1993. Eesti keele grammatika II. Süntaks [Estonian grammar II. Syntax]. Tallinn: Eesti TA Keele ja Kirjanduse instituut.

Forcada, Mikel L., Mirela Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez and Francis M. Tyers. 2011. Apertium: A free/open-source platform for rule-based machine translation. Machine Translation 25. 127–144.

Hasselblatt, Cornelius. 1990. Das Estnische Partikelverb Als Lehnübersetzung Aus Dem Deutschen (Veroffentlichungen Der Societas Uralo-Altaica). Otto Harrassowitz: Wiesbaden.

Kaalep, Heiki-Jaan. 1997. An Estonian morphological analyser and the impact of a corpus on its development. Computers and the Humanities 31. 115–133.

Kaalep, Heiki-Jaan and Tarmo Vaino. 2001. Complete morphological analysis in the linguist's toolbox. In Proceedings of Congressus Nonus Internationalis Fenno-Ugristarum, Pars V. 9–16.

[13] https://korpused.keeleressursid.ee/syntaks

[14] https://github.com/EstSyntax/EstCG

Karlsson, Fred, Arto Anttila, Juha Heikkilä and Atro Voutilainen. 1995. Constraint grammar: A language-independent system for parsing unrestricted text. Berlin & New York: Mouton de Gruyter.

Lindström, Liina and Kaili Müürisep. 2009. Parsing corpus of Estonian dialects. In Proceedings of Workshop on Constraint Grammar and robust parsing at NODALIDA 2009 (NEALT Proceedings 8).

McDonald, Ryan, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. 92–97.

Muischnek, Kadri, Kaili Müürisep and Tiina Puolakainen. 2013. Estonian particle verbs and their syntactic analysis. In Z. Vetulani and H. Uszkoreit (eds.) Human language technologies as a challenge for computer science and linguistics: 6th Language & Technology Conference proceedings. Poznań: Adam Mickiewicz University. 338–342.

Muischnek, Kadri, Kaili Müürisep and Tiina Puolakainen. 2014a. Dependency parsing of Estonian: Statistical and rule-based approaches. In Human language technologies – The Baltic perspective: The Sixth International Conference "Human Language Technologies – The Baltic Perspective". Amsterdam: IOS Press. 111–118.

Muischnek, Kadri, Kaili Müürisep and Tiina Puolakainen. 2016. Estonian dependency treebank: From constraint grammar tagset to universal dependencies. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).

Muischnek, Kadri, Kaili Müürisep, Tiina Puolakainen, Eleri Aedmaa, Riin Kirt and Dage Särg. 2014b. Estonian dependency treebank and its annotation scheme. In Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13). 285–291.

Müürisep, Kaili. 2000. Eesti keele arvutigrammatika: süntaks. [Computational grammar of Estonian: Syntax]. Doctoral dissertation. University of Tartu.

Müürisep, Kaili and Helen Nigol. 2008. Where do parsing errors come from: The case of spoken Estonian. In P. Sojka, A. Horák, I. Kopečekp and K. Pala (eds.) Proceedings of Text, Speech and Dialogue. Berlin: Springer. 161–168.

Müürisep, Kaili and Helen Nigol. 2009. Shallow parsing of transcribed speech of Estonian and disfluency detection. In Z. Vetulani and H. Uszkoreit (eds.) Human language technology: Challenges of information society. Berlin: Springer. 165–177.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13. 95–135.

Orasmaa, Siim, Timo Petmanson, Alexander Tkachenko, Sven Laur and Heiki-Jaan Kaalep. 2016. EstNLTK - NLP Toolkit for Estonian. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).

Puolakainen, Tiina. 2001. Eesti keele arvutigrammatika: morfoloogiline ühestamine [Computational grammar of Estonian: Morphological disambiguation]. Doctoral dissertation. University of Tartu.

Rosa, Rudolf, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman and Zdeněk Žabokrtský. 2014. HamleDT 2.0: Thirty dependency treebanks Stanfordized. In Proceedings of LREC 2014.

Särg, Dage. 2015. Adapting constraint grammar for parsing Estonian chatroom texts. In M. Dickinson, E. Hinrichs, A. Patejuk and A. Przepiórkowski (eds.) Proceedings of the 14th International Workshop on Treebanks and Linguistic Theories (TLT14). Warsaw: Polish Academy of Sciences. 300–307.

Uibo, Heli, Jaak Pruulmann-Vengerfeldt, Jack Rueter and Sulev Iva. 2015. Oahpa! õpi! opiq! Developing free online programs for learning estonian and võro. In Proceedings of the 4th Workshop on NLP for Computer Assisted Language Learning at NODALIDA 2015 (NEALT Proceedings 26). 51–64.

Villavicencio, Aline and Ann Copestake. 2002. Verb-particle constructions in a computational grammar of English. In Proceedings of the Ninth International Conference on Head-Driven Phrase Structure Grammar. Stanford: CSLI Publications.