Parsing Estonian: Tools and Resources

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

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

Tiina Puolakainen
University of Tartu
tiina.puolakainen@ut.ee

Krista Liin
University of Tartu
krista.liin@ut.ee

January 3, 2016

Abstract

This article gives an overview of the state of the art of tools and resources for syntactic analysis of Estonian. A morphosyntactic disambiguator, surface-syntactic analyser and dependency parser are all based on the Constraint Grammar formalism. Also, the paper describes some experiments conducted with the statistical parser. As for language resources, a 400,000-word manually annotated dependency treebank has been created. Our tools have also been tested by large-scale corpus annotation.

1 Introduction

This paper describes a set of tools and resources for parsing Estonian texts starting from morphological analysis and disambiguation to dependency parsing and syntax-based applications. In 1995, the first version of morphological analyser of Estonian ESTMORF was created and already couple of years later it was able to assign adequate morphological descriptions to 99% tokens in text [1]. In the same year, Fred Karlsson together with his colleagues published a monograph on Constraint Grammar [2], a framework for disambiguating and parsing non-restricted text that has been successfully used not only for analysing the Indo-European languages but also e.g. for analysing Finnish.

That spurred the work on Estonian Constraint Grammar (EstCG). Its earlier versions used a locally developed parsing engine, but its last version uses VISL CG-3 format and software [3]. EstCG parser consists of separate sets of grammar rules for determining clause boundaries, morphological disambiguation, surface syntactic analysis and determining dependency relations. In addition to rule-sets, the system also includes several valency lexicons and a special module for identifying particle verbs [4].

The rest of the paper is organized as follows. Sections 2 and 3 provide an overview of morphological disambiguation and clause boundary detection module, sections 4 and 5 describe the grammar of surface and dependency syntax, section 6 reports the experimental results of applying MaltParser to Estonian. Section 7 gives an overview of graphical user interface for combining different modules together. In section 8, we describe Estonian Dependency Treebank. In section 9, we conclude the paper with describing some applications of our syntactic tools and discussing some ideas for future work.

2 Morphosyntactic disambiguator

EstCG parser takes morphologically analysed text as input, i.e. each word-form has all the possible morphological analysis attached to it. Morphological ambiguity rate of Estonian text is ca 50%. For example word-form *või* can be noun *või* ('butter') in nominative or genitive case-form, negative present tense form of verb *võima* ('may') in all three persons in singular and plural or conjunction *või* ('or').

Constraint Grammar rules for morphological disambiguation delete the readings that are inappropriate regarding the context. Preliminary clause boundaries are also set at the same stage. If it is not possible to disambiguate basing on the contextual information, all possible readings are retained.

The disambiguating grammar consists of more than 3400 handwritten rules, almost a quarter of them address certain word-forms. For example a very frequent word-form *on* is ambiguous between the readings of simple present 3rd person singular and plural of the verb *olema* ('to be'). The other rules can again cover broader ambiguity classes.

A difficult case for disambiguation is the choice between readings of nominative, genitive, partitive or short illative (additive) case forms of a noun. This type of ambiguity tends to be more characteristic of frequent and common words, eg. nouns *ema* ('mother') and *isa* ('father') are ambiguous between nominative, genitive and partitive readings. The word-form *metsa* 'forest' is an example of typical homonymous form of singular genitive, partitive and short illative cases. Its parallel form of illative case, in this example *metsasse* ('into the forest'), is actually not used in Estonian.

The other frequent sources of errors and ambiguities are participles (they are always four-way ambiguous: negative indicative past tense, past participle, adjectival use of past participle and noun as a nominalisation of an adjective), and also ambiguous readings of adposition, adverb and noun of some word forms.

For example, *peale* can be an autonomous adverb (most general meaning 'onto') or a particle as a part of a particle verb, e.g. *peale sattuma* 'stumble on/across'; it can be also a postposition governing a noun in genitive case (meaning 'in addition to') or elative case (meaning 'starting from') or preposition governing a noun in genitive case or partitive case (meaning 'after'); after all, *peale* can be also a noun *pea* ('head') in a singular allative case. As a consequence of this multi-way ambiguity of the word-form *peale*, the Estonian phrase *asetama selle peale* can have 3 different meanings: (1) 'to place onto this' with *peale* as a postposition; (2) 'to place this onto' with *peale* as a particle; (3) 'to place this onto the head'.

Tests made with a 26,700 word test corpus showed results of 97.1% recall and 90.2% precision. In other words, the output contained 2.9% of errors (word-forms that did not contain the correct reading among all survived readings in a cohort) and 9.8% of retained readings were not correct (superfluous). The initial morphologically analysed text contained 51.8% of superfluous readings and 0.6% of word-forms did not have a correct reading in the cohort (recall 99.4% and precision 48.2%). This happens most often with unknown words, mostly proper noun, but also in other cases, for example word-form *väikesed* ('small') is given only an adjective reading but in some sentences it is functioning clearly as a noun.

A common source of errors are elliptical sentences as for example a title *Suhtlemise puudus* ('Lack of communication'), there the word-form *puudus* ('lack') is considered to be a verb being an only possibility for that in the sentence, but in this case should be a noun as the sentence contains only one noun phrase.

One of the hardest tasks is disambiguation of noun forms with homonymous nominative, genitive and partitive or genitive, partitive and additive case forms. The following sentence (1) has two appropriate readings depending what role the noun *rõõm* ('joy') is playing in the sentence – an object of the main verb and consequently has to be considered being in partitive case or a modifier of a noun *koostegemine* ('cooperation') and then accordingly in genitive case:

(1) a. *Külades* *tuntakse* *rõõmu* *koostegemisest.*
village[INE.PL] know[IMPS.PL] joy[GEN] cooperation[ELA] (INE=inessive)
'The cooperation of joy is known in the villages.' (ELA=elative)

b.	*Külades*	*tuntakse*	*rõõmu*	*koostegemisest.*	
	village[INE.PL]	feel[IMPS.PL]	joy[PAR]	cooperation[ELA]	(PAR=partitive)

‘There is a feeling of joy of cooperation in the villages.’

3 Clause boundary detector

The clause boundary annotation is a simple way to constrain context of morphosyntactic rules, also, the performance of statistical parser improved if the model had information about clause boundaries. Currently, EstCG has ca 80 hand-crafted rules for detecting clause boundaries. The beginning of each clause is annotated by a special label.

The rules mainly consider 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 clause-initial position from coordinating or modifying usage within a clause, as a morphologically analysed (but not yet disambiguated) text contains plenty of ambiguities for different interpretations (a classical but not single example is past participles that can function as a predicate of a clause or just an adjectival modifier of a noun). Also special clause boundary tags are introduced for embedded clauses, where, for example, a subject and a predicate of main clause may be separated by a relative clause and therefore would be not related to each other without special care.

4 Surface-oriented syntactic analyser

The syntactic or, more precisely, the surface-syntactic module of the EstCG adds a label of syntactic function to every word-form in the text. According to the EstCG annotation scheme, the members of the verbal chain can be finite or infinite main verbs (FMV, IMV), and finite or infinite auxiliaries (FCV, ICV). Also, we distinguish particles as parts of particle verb (VPart), and verb negators (NEG). The arguments of the verb are labelled as subject (SUBJ), object (OBJ), predicative (PRD) or adverbial (ADVL); the adjuncts also get the adverbial label. The attributes of a nominal are tagged according to their part-of-speech (AN for adjectives, NN for nouns, KN for adpositions, DN for adverbs and INFN - for infinitives). We distinguish the nouns governed by an adposition with a special label (<P or P>) and also nouns governed by a quantifier (<Q or Q>). There is a special symbol for indicating whether the word form is a pre- or postmodifier (<NN or NN> for example). Also, we label direct addresses (VOC), conjunctions (J) and interjections (I).

The annotation created by the analyser is very shallow: the clause boundaries are set and the syntactic functions of the word-forms in every clause are labelled, but no inter-clausal relations are identified.

Also, the head verbs are not connected with their arguments. For example, if a clause contains an infinitive subclause and both verbs have an object, there is no way to tell from the annotation which object complements which verb. Also, the objects are not connected with their head verbs and if a clause contains an infinitive subclause and both verbs having an object, there is no way to tell from the annotation which object complements which verb. There is no direct connection between an attribute and its head, but pre- and postmodifying attributes are distinguished.

The adverbials form a large and heterogeneous class, also sentence and phrase adverbials are not distinguished. So both word-forms *väga* ('very') and *kiiresti* ('quickly') get the label ADVL in the sentence *Ta jooksis väga kiiresti* ('S/he ran very quickly').

Deeper syntactic analysis is the goal of the next grammar module, a module for building dependency trees.

During the surface syntactic analysis, first all possible labels are added depending on the part-of-speech tag and grammatical categories. 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 the 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 is a verb that under certain circumstances takes only a total object[1] (i.e. an object in genitive or nominative case) or if the same clause contains a noun with non-ambiguous object reading and the word-form under consideration is not in a coordinating relation with that.

The module for surface syntactic analysis comprises ca. 1300 rules. Experiments on a manually annotated 9500-token corpus showed that the recall of the whole syntactic analysis (including morphological disambiguation) was 92.9% and precision 69.3%; the error rate was 7.1%. It means that 7.1% of tokens don't get the correct label and 30–31% of the added labels are either superfluous or erroneous.

The majority of errors occur in annotating objects, subjects and predicatives 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, a modifier of a quantifier. Also, the nouns in

[1] 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 [5, pages 96–97].

nominative, genitive or partitive case can act as adverbials, appositions, belong to the adposition phrase or perform some less observed roles in the sentence.

A substantial amount of non-solved ambiguity in the output is caused by the indiscernibility of adverbials and adverbial attributes. The problem is similar to pp-attachment, e.g. in the sentence *Seal tuleb mees metsast* ('There comes a man from forest') the word-form *metsast* ('from forest') is ambiguous between adverbial and attributive readings.

5 Dependency parser and particle verb detector

Recently, the EstCG parser has been enhanced with dependency rules and this stage is still under development. However, the analysis provided by CG dependency parser helped to develop the first version of Estonian Dependency Treebank, consisting of 400,000 words [6], which in turn gave an opportunity to experiment with statistical parsing methods, namely training and evaluating MaltParser [7] for analysing Estonian texts.

The grammar of dependencies consists of ca 600 rules. The EstCG parser achieves an unlabeled attachment score (UAS) of 77.2%.

We added a special module of rules in order to recognize particle verbs i.e. multiword expressions consisting of a verb and an adverbial particle, also called phrasal verbs in more general terms. The module for identification of Estonian particle verbs consists of a grammar of approximately 500 rules and a thorough lexicon for 70 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 particle verbs receive correct analysis at the shallow syntactic level and 95–100% of 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.

In the following example (see Figure 1) there are two different correct translations of the sentence depending on the choice of taking the *üle* ('over') as a preposition (2a) or as a part of a particle verb *üle mängima* ('to outplay') (2b):

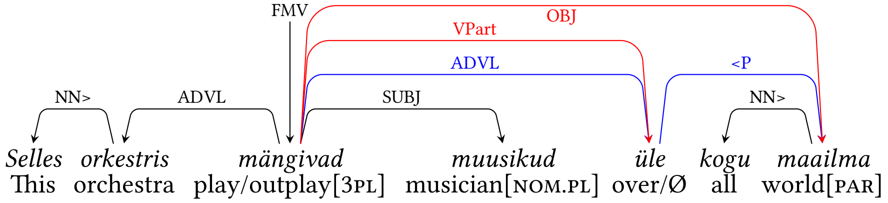

FMV
OBJ
VPart
ADVL
<P
NN>
ADVL
SUBJ
NN>
Selles orkestris mängivad muusikud üle kogu maailma
This orchestra play/outplay[3PL] musician[NOM.PL] over/Ø all world[PAR]

Figure 1: Two alternative analyses of the sample sentence.

(2) a. ‘The musicians around the world are playing in this orchestra.’

b. ‘The musicians are outplaying all the world in this orchestra.’

6 Statistical parser

For our first experiments with statistical analysis we have selected MaltParser [7] 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 [8] which helps the end user to select the appropriate parameters and parsing algorithm without having expert knowledge on underlying methods.

First, we transformed the texts from the CG format to the 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 (e.g. intransitivity of verbs). The syntactic labels remain 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.

Only the part of the treebank that was double-checked at that point of time (191,000 tokens, 13,310 sentences) was used for statistical parsing. Half of the corpus consists of newspaper texts, while the other half contains fiction and scientific texts. All the sentences have been manually morphologically disambiguated. Every 5th sentence was moved to the testing part of corpora, so the training set consisted of 153,471 tokens. 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 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.

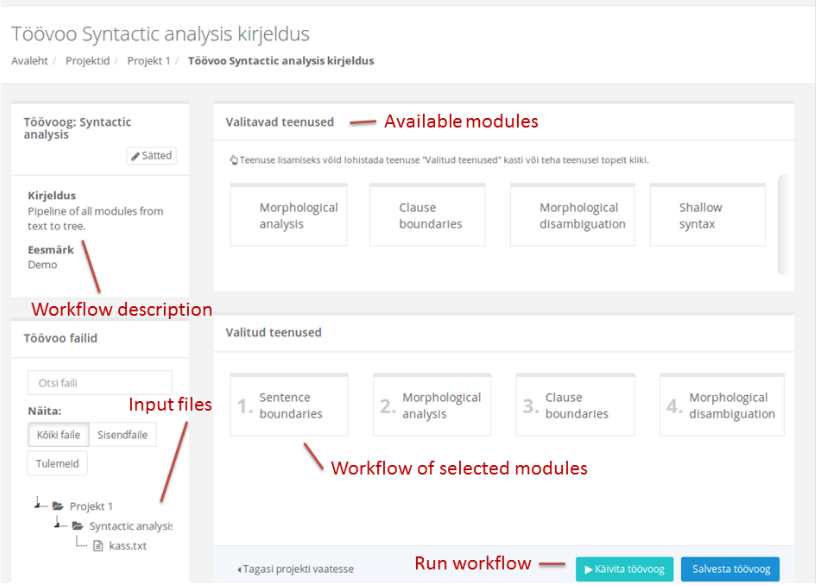
After excluding punctuation marks and non-sentential constructions from the analysis, the LAS decreased to 80.3% (31,434 tokens). Also, we observed the unlabeled attachment score (UAS) of 83.4% and the label accuracy (LA) of 88.6%.

We have conducted several experiments on running Maltparser along with EstCG parser: using syntactic information provided by EstCG parser as input for Maltparser or applying special fixing rules to the output of Maltparser. These improved overall performance by 1% [6].

7 Language pipeline

In order to make language technology easier to use for people who are not at home in the command line programs, there is also a graphical web interface for executing annotation workflows - Keeleliin[2] (Language Pipeline). In this interface, it is possible to combine different modules, such as morphological disambiguation or dependency annotation (e.g. picking either Constraint grammar or MaltParser) into reusable workflows that are then executed in the server (see Figure 2). It is also possible to share prepared workflows with other users, so that users with little knowledge about the underlying structure can also use Keeleliin to annotate their texts with different syntactic workflows with no need to install anything beforehand.

At the moment Keeleliin is still in development, so the majority of modules will not be inserted until 2016, as the respective web services are made available. The current version is already open for testing to academic users.


Töövoo Syntactic analysis kirjeldus

Avaleht / Projektid / Projekt 1 / Töövoo Syntactic analysis kirjeldus

Töövoog: Syntactic analysis

Kirjeldus
Pipeline of all modules from text to tree.

Eesmärk
Demo

Valitavad teenused

Teenuse lisamiseks võid lohistada teenuse "Valitud teenused" kasti või teha teenusel topelt kliki.

Valitud teenused

1. Sentence boundaries
2. Morphological analysis
3. Clause boundaries
4. Morphological disambiguation

Figure 2: Creating a workflow in the language pipe web service. View of available modules is restricted to those that accept the output format of already selected modules.

[2] http://keeleliin.keeleressursid.ee.

8 Corpora and treebanks

The initial versions of the EstCG parser were developed basing on the linguistic knowledge as presented in a descriptive grammar of Estonian [9] 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 a machine learning based parser, creating a larger manually annotated corpus was essential.

We succeeded to get funding for creating an Estonian Dependency Treebank and completed its first version by the end of 2014 [10]. The treebank contains approximately 400,000 tokens and is annotated for part of speech, morphological description, syntactic functions and dependency relations.[3]

Figure 3 depicts an Estonian sentence *Hommikul püüdis kass kinni kena paksu hiire* ('In the morning, the cat caught a nice fat mouse'). For every word in the sentence there is a separate row for its analysis. It begins with a lemma, followed by an inflectional ending, POS tag and morphological description.[4] The syntactic function labels begin with @ and tags indicating dependency relations with #.

```
"<s>"
"<Hommikul>"
     "hommik" L1 S com sg ad cap @ADVL #1->2                morning
"<püüdis>"
    "püüd" Lis V main indic impf ps3 sg ps af @FMV #2->0    caught
"<kass>"
    "kass" L0 S com sg nom @SUBJ #3->2                      cat
"<kinni>"
    "kinni" L0 D @Vpart #4->2                               verbal particle
"<kena>"
    "kena" L0 A pos sg gen @AN> #5->7                       nice
"<paksu>"
    "paks" L0 A pos sg gen @AN> #6->7                       fat
"<hiire>"
    "hiir" L0 S com sg gen @OBJ #7->2                       mouse
"<.>"
    "." Z Fst CLB #8->7
"</s>"
```

Figure 3: Sample sentence "In the morning, the cat caught a nice fat mouse."

In order to join in an international effort and to make available the Estonian Dependency Treebank with a cross-linguistically consistent treebank annotation for many languages we have started with conversion of the afforementioned treebank to the Universal Dependencies [11] annotation scheme.[5]

Perhaps there is no better method to test a program for linguistic analysis than

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

[4] explained in detail in http://www.cl.ut.ee/korpused/morfliides/seletus.php?lang=en.

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

large-scale corpus annotation; at least we decided to test our tools this way.

There exists a relatively big corpus of contemporary Estonian.[6] A subcorpus of the afforementioned big corpus (Balanced Corpus, 15 million tokens) was parsed using the CG surface-syntax rules. Resulting language resource is available in two ways: one can query the corpus using corpus query interface at Keeleveeb[7] or one can obtain the full parsed corpus at request.

9 Conclusions and future work

The plans for the near future include experiments for combining rule-based CG parser and MaltParser and also experimenting with other statistical parsers, e.g. Mate [12] or LingPars [13].

We have already started converting the Estonian Dependency Treebank to Universal Dependencies annotation scheme.

Building a morphosyntactic and syntactic analyser 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 [14, 15] – programs using linguistic tools for generating new tasks for language learner and testing the student's 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 the student's answer. Estonian Oahpa! and Vasta! are currently under development [16]. Another system where we are planning to employ Estonian Constraint Grammar is rule-based machine translation platform Apertium [17].

One can test our demo version of the syntactic parser at `https://korpused.keeleressursid.ee/syntaks` or install it as an open-source software.[8]

Acknowledgements

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

[6] `http://www.cl.ut.ee/korpused/segakorpus/`.

[7] `http://www.keeleveeb.ee`.

[8] `https://github.com/EstSyntax/EstCG`

References

[1] Heiki Jaan Kaalep. An Estonian Morphological Analyser and the Impact of a Corpus on Its Development. *Computers and the Humanities*, 31(2):115–133, 1997.

[2] Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text.* Mouton de Gruyter, Berlin, 1995.

[3] Eckhard Bick and Tino Didriksen. CG3 Beyond Classical Constraint Grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics NODALIDA 2015*, pages 31–40, 2015.

[4] Kadri Muischnek, Kaili Müürisep, and Tiina Puolakainen. Estonian particle verbs and their syntactic analysis. In Z. Vetulani and H. Uszkoreit, editors, *Human Language Technologies as a Challenge for Computer Science and Linguistics: 6Th Language & Technology Conference Proceedings*, pages 338–342. Adam Mickiewicz University, 2013.

[5] Mati Erelt, editor. *Estonian Language*, volume 1 of *Linguistica Uralica Supplementary series.* 2003.

[6] Kadri Muischnek, Kaili Müürisep, and Tiina Puolakainen. Dependency Parsing of Estonian: Statistical and Rule-based Approaches. In Andrius Utka, Gintare Grigonyte, Jurgita Kapociute-Dzikiene, and Jurgita Vaicenoniene, editors, *Baltic HLT*, volume 268 of *Frontiers in Artificial Intelligence and Applications*, pages 111–118. IOS Press, 2014.

[7] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[8] Miguel Ballesteros and Joakim Nivre. MaltOptimizer: An Optimization Tool for MaltParser. In Walter Daelemans, Mirella Lapata, and Lluís Màrquez, editors, *EACL*, pages 58–62. The Association for Computer Linguistics, 2012.

[9] M. Erelt, R. Kasik, H. Metslang, H. Rajandi, K. Ross, H. Saari, K. Tael, and S. Vare. *Eesti keele grammatika II. Süntaks.* Eesti TA Keele ja Kirjanduse instituut, 1993.

[10] Kadri Muischnek, Kaili Müürisep, Tiina Puolakainen, Eleri Aedmaa, Riin Kirt, and Dage Särg. Estonian Dependency Treebank and its annotation scheme. In

Verena Henrich et al., editors, *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*, pages 285–291. University of Tübingen, 2014.

[11] Ryan McDonald, 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. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 92–97, 2013.

[12] Bernd Bohnet. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING*, pages 89–97. Tsinghua University Press, 2010.

[13] Eckhard Bick. LingPars, a Linguistically Inspired, Language-Independent Machine Learner for Dependency Treebanks. In Lluís Màrquez and Dan Klein, editors, *CoNLL*, pages 171–175. ACL, 2006.

[14] Lene Antonsen, Saara Huhmarniemi, and Trond Trosterud. Interactive pedagogical programs based on constraint grammar. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, volume 4 of *NEALT Proceedings Series*, pages 10–17, 2009.

[15] Lene Antonsen, Saara Huhmarniemi, and Trond Trosterud. Constraint Grammar in Dialogue Systems. In *Proceedings of the NODALIDA 2009 workshop Constraint Grammar and robust parsing*, volume 8 of *NEALT Proceedings Series*, pages 13–21, 2009.

[16] Heli Uibo, Jaak Pruulmann-Vengerfeldt, Jack Rueter, and Sulev Iva. 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*, volume 26 of *NEALT Proceedings Series*, pages 51–64. Linköping University Electronic Press, 2015.

[17] Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144, 2011.